(corresponding to US 7,461,249 B1)

(14)

(19)日本国特許庁（ＪＰ）　(12) **公表特許公報（Ａ）**　(11)特許出願公表番号
特表2003－507785
（P2003－507785A）
(43)公表日　平成15年２月25日(2003.2.25)

(51)Int.Cl.'	識別記号	FI	テーマコード'(参考)
G06F 1/00		G06F 12/14	320A 5B017
12/14	320	9/06	660A 5B076
			660H

審査請求　未請求　予備審査請求　有　（全119頁）

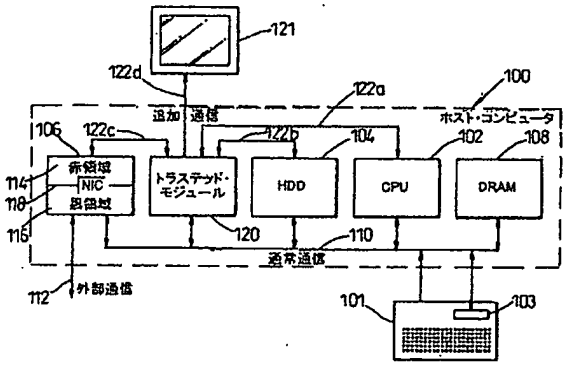
(21)出願番号　特願2001－517235(P2001－517235)
(86)(22)出願日　平成12年８月11日(2000.8.11)
(85)翻訳文提出日　平成14年２月13日(2002.2.13)
(86)国際出願番号　ＰＣＴ／ＧＢ００／０３１０１
(87)国際公開番号　ＷＯ０１／０１３１９９
(87)国際公開日　平成13年２月22日(2001.2.22)
(31)優先権主張番号　９９３０６４１５．３
(32)優先日　平成11年８月13日(1999.8.13)
(33)優先権主張国　欧州特許庁（ＥＰ）
(81)指定国　ＥＰ(ＡＴ，ＢＥ，ＣＨ，ＣＹ，ＤＥ，ＤＫ，ＥＳ，ＦＩ，ＦＲ，ＧＢ，ＧＲ，ＩＥ，ＩＴ，ＬＵ，ＭＣ，ＮＬ，ＰＴ，ＳＥ)，ＪＰ，ＵＳ

(71)出願人　ヒューレット・パッカード・カンパニー
ＨＥＷＬＥＴＴ－ＰＡＣＫＡＲＤ　ＣＯＭＰＡＮＹ
アメリカ合衆国カリフォルニア州パロアルト　ハノーバー・ストリート　3000
(72)発明者　ピアソン，シアニ，ライン
イギリス国ブリストル・ビーエス９・３ピーゼット，ウエストバーリィーオン－トリム，サンディリーズ・35
(74)代理人　弁理士　古谷　馨　（外３名）

最終頁に続く

(54)【発明の名称】　コンピュータ・プラットフォームおよびその運用方法

(57)【要約】
コンピュータ・プラットフォーム(100)は、コンピュータ・プラットフォームの不正防止コンポーネント(120)またはトラステッド・モジュールをソフトウェアと共に使用する。このソフトウェアは、前記不正防止コンポーネント内で動作することが好ましく、そのプラットフォーム上のデータのアップロード及び使用を、そのプラットフォームの汎用ドングルとして制御する。ライセンス検査はトラステッド環境（即ち、ユーザが期待するように振舞うことが信用できる環境）下で行なわれる。これは、ソフトウェアのアップロードとライセンス検査を完全性検査することにより強制することができる。測定記録は不正防止装置に格納され、要求に応じて管理者へ報告される。データについての登録及び支払いを可能にするための、関連するクリアリングハウス機構が存在してもよい。



【特許請求の範囲】

【請求項１】

内部の不正に対して抵抗力があり、第三者の公開鍵証明書を格納するトラステッド・モジュールと、

プラットフォームまたは該プラットフォームのユーザが特定のデータの使用を許諾されているか否かを検査し、該データを使用するための、及び、該データの使用を監視するための、あるいはその両方のためのインタフェースを提供するセキュア実行プログラムと、前記プラットフォームまたは該プラットフォームのユーザが特定のデータのインストールを許諾されているか否かを検査するための、及び、インストール前にデータの完全性を検査するための、あるいはその両方を行うためのセキュア・ローダとのうち、少なくとも１つを含むライセンス関連コードを格納する手段と、

第三者の秘密鍵を用いて署名されたライセンス関連コードのハッシュされたバージョンを格納する手段とから成り、

前記プラットフォームのブート時において、前記ライセンス関連コードは、前記署名されたバージョンと前記公開鍵証明書を参照して完全性検査が成され、前記完全性検査が失敗した場合、前記ライセンス関連コードはロードされないようにプログラムされているコンピュータ・プラットフォーム。

【請求項２】

前記完全性検査は、前記ライセンス関連コードを読み込み、及び、ハッシュして第１のハッシュを生成し、前記公開鍵証明書を用いて前記署名されたバージョンを読み込み、及び、復号して第２のハッシュを生成し、前記第１および第２のハッシュを比較することにより実施される、請求項１のコンピュータ・プラットフォーム。

【請求項３】

前記ライセンス関連コードは、前記トラステッド・モジュールと他のコンピュータ・プラットフォームのさらなるトラステッド・モジュールとの間でライセンス鍵の転送を可能にするためのセキュア鍵転送コードをさらに含む、請求項１または２のコンピュータ・プラットフォーム。

【請求項４】

前記ライセンス関連コードは、前記トラステッド・モジュールと通信するために呼び出されることが可能なインタフェース・サブルーチンのライブラリをさらに含む、請求項１～４のうちのいずれか１項に記載のコンピュータ・プラットフォーム。

【請求項５】

前記ライセンス関連コードは、前記ライセンス関連コードは、データの少なくとも１グループに対して、前記データのグループのそれぞれを指定し、前記データのグループへのインタフェースとして機能することの可能なソフトウェア実行プログラムを含む、請求項１～４のうちのいずれか１項に記載のコンピュータ・プラットフォーム。

【請求項６】

前記ライセンス関連コードを格納する前記手段、及び、前記ライセンス関連コードの前記ハッシュされたバージョンを格納する前記手段、または、これら両方は、前記トラステッド・モジュールによって、少なくとも一部に設けられる、請求項１～５のうちのいずれか１項に記載のコンピュータ・プラットフォーム。

【請求項７】

前記トラステッド・モジュール及び前記プラットフォームのオペレーティング・システムは、これらの間に前記コンピュータ・プラットフォームの他の部分へアクセスできない専用通信路を有する、請求項１～６のうちのいずれか１項に記載のコンピュータ・プラットフォーム。

【請求項８】

前記オペレーティング・システムは、前記プラットフォームまたは前記プラットフォームのユーザが前記特定のデータをインストールすることを許諾されているか否かについてライセンス検査すること、及び、前記特定データの完全性を検査すること、または、これら両方を行うことを前記セキュア・ローダへ要求するように動作可能であり、

前記要求に応答して、前記セキュア・ローダは、前記検査を実施し、前記検査の結果を前記オペレーティング・システムへ応答するように動作可能であり、

前記応答に応じて、前記オペレーティング・システムは、前記特定のデータをインストールする、または、インストールしないように動作可能である、請求項１～７のうちのいずれか１項に記載のコンピュータ・プラットフォーム。

【請求項９】

前記オペレーティング・システムは、前記セキュア・ローダのみに応答して、前記特定のデータをインストールするようにプログラムされている請求項８のコンピュータ・プラットフォーム。

【請求項１０】

前記トラステッド・モジュールは、インストールされる前記特定のデータと関連するパーティの公開鍵証明書を格納し、

前記オペレーティング・システムは、前記検査の要求において、前記関連するパーティの秘密鍵を用いて署名された前記特定データのハッシュ・バージョンと共に前記特定データを含めるように動作可能であり、

前記検査の実施において、前記セキュア・ローダは、前記要求に含まれる前記特定データをハッシュして第３のハッシュを生成し、前記関連するパーティの公開鍵証明書を用いて前記要求の署名されハッシュされたバージョンを復号して第４のハッシュを生成し、前記第３及び第４のハッシュが合致するか否かに応じて前記応答を生成するように動作可能である、請求項８または９のコンピュータ・プラットフォーム。

【請求項１１】

前記検査の要求は、前記特定のデータのための前記ソフトウェア実行プログラムを含む、請求項５に直接的または間接的に従属する請求項１０のコンピュータ・プラットフォーム。

【請求項１２】

前記ソフトウェア実行プログラム（または前記ソフトウェア実行プログラムのうちの少なくとも１つ）は、特定のデータをインストールすることを前記トラステッド・モジュールへ要求するように動作可能であり、

前記要求に応答して、前記トラステッド・モジュール内の前記セキュア・ローダは、前記プラットフォームまたは前記プラットフォームのユーザが前記特定の

データをインストールすることを許諾されているか否かについてライセンス検査すること、及び、前記データの完全性を検査すること、または、これら両方を行い、前記検査の結果を前記オペレーティング・システムへ応答するように動作可能であり、

前記応答に応じて、前記オペレーティング・システムは、前記特定のデータをインストールする、または、インストールしないように動作可能である、請求項５に従属する請求項６、または、これに従属する請求項７～１１のうちいずれか１項に記載のコンピュータ・プラットフォーム。

【請求項１３】

前記オペレーティング・システムは、前記トラステッド・モジュールにのみ応答して、前記特定のデータをインストールするようにプログラムされている、請求項１２のコンピュータ・プラットフォーム。

【請求項１４】

前記トラステッド・モジュールから前記オペレーティング・システムへの前記応答は、前記専用通信路を介して供給される、請求項７に従属する請求項１２または１３のコンピュータ・プラットフォーム。

【請求項１５】

前記検査が成功した場合、前記トラステッド・モジュールは、前記特定のデータを監査するためにログを生成するように動作可能である、請求項８～１４のうちのいずれか１項に記載のコンピュータ・プラットフォーム。

【請求項１６】

前記検査が成功した場合、前記セキュア・ローダは、前記特定のデータに対してウィルス検査を行うように動作可能である、請求項８～１５のうちのいずれか１項に記載のプラットフォーム。

【請求項１７】

インストール時において、前記特定のデータは前記トラステッド・モジュールへインストールされる、請求項８～１６のうちいずれか１項に記載のコンピュータ・プラットフォーム。

【請求項１８】

さらなる着脱可能なトラステッド・モジュールをさらに含み、

第１のトラステッド・モジュールと前記着脱可能なトラステッド・モジュールとの間で認証検査を実施するように動作可能であり、

インストール時に、前記特定のデータは、前記さらなるトラステッド・モジュールへインストールされる、請求項８～１６のうちいずれか１項に記載のコンピュータ・プラットフォーム。

【請求項１９】

前記ソフトウェア実行プログラム（または前記ソフトウェア実行プログラムのうちの少なくとも１つ）は、前記トラステッド・モジュールの公開鍵と前記データのそれぞれに対するライセンスモデルとを含み、

前記オペレーティング・システムは、前記ソフトウェア実行プログラムのそれぞれのデータが使用されることを、そのソフトウェア実行プログラムへ要求するように動作可能であり、

前記要求に応答して、前記ソフトウェア実行プログラムは、前記ソフトウェア実行プログラムのライセンスモデルを用いて、前記プラットフォームまたは前記プラットフォームのユーザが前記データの使用を許諾されているか否かをライセンス検査するように、前記セキュア実行プログラムへ要求するように動作可能であり、

後者の要求に応答して、前記セキュア実行プログラムは、前記要求されたライセンス検査を実施し、前記トラステッド・モジュールの秘密鍵を用いて前記ライセンス検査の結果に署名し、前記署名された結果を前記ソフトウェア実行プログラムへ応答するように動作可能であり、

前記応答に応答して、前記ソフトウェア実行プログラムは、前記トラステッド・モジュールの公開鍵を用いて署名された結果の完全性を検査し、前記ライセンス検査が成功した結果前記完全性検査が成功すると、前記データの使用を前記オペレーティング・システムへ要求するように動作可能である、

請求項５、または、これに直接的または間接的に従属する請求項６～１８のうちのいずれか１項に記載のコンピュータ・プラットフォーム。

【請求項２０】

前記ソフトウェア実行プログラム（または前記ソフトウェア実行プログラムのうちの少なくとも１つ）は、前記トラステッド・モジュールの公開鍵と前記データのそれぞれのライセンスモデルを含み、

前記オペレーティング・システムは、前記特定のデータが使用されることを、前記セキュア実行プログラムへ要求するように動作可能であり、

前記要求に応答して、前記セキュア実行プログラムは、前記特定のデータに対するライセンスモデルに対する、前記トラステッド・モジュールの秘密鍵を使用して署名された要求を、前記それぞれのソフトウェア実行プログラムへ送信するように動作可能であり、

後者の要求に応答して、前記ソフトウェア実行プログラムは、前記トラステッド・モジュールの前記公開鍵を用いて前記要求の完全性を検査し、完全性検査が成功すると、前記ライセンスモデルを前記セキュア実行プログラムへ送信するように動作可能であり、

前記ライセンスモデルの受信すると、前記セキュア実行プログラムは、前記ライセンスモデルを用いてライセンス検査を実施し、ライセンス検査が成功すると、前記データを使用することを前記オペレーティング・システムへ要求するように動作可能である、

請求項５、またはこれに直接的または間接的に従属する請求項６～１９のうちいずれか１項に記載のコンピュータ・プラットフォーム。

【請求項２１】

前記セキュア実行プログラムは、少なくとも１つのライセンスモデルを含み、

前記オペレーティング・システムは、前記特定のデータが使用されることを、前記セキュア実行プログラムへ要求するように動作可能であり、

前記要求に応答して、前記セキュア実行プログラムは、前記ライセンスモデルまたは前記ライセンスモデルのうちの１つを用いてライセンス検査を実施し、ライセンス検査が成功すると、前記データを使用することを前記オペレーティング・システムへ要求するように動作可能である、

請求項の１～２０のうちのいずれか１項に記載のコンピュータ・プラットフォーム。

【請求項２２】

前記オペレーティング・システムは、前記セキュア実行プログラムまたは前記ソフトウェア実行プログラムにのみ応答して、前記特定のデータをインストールするようにプログラムされている、請求項１９～２１のうちいずれか１項に記載のコンピュータ・プラットフォーム。

【請求項２３】

前記セキュア実行プログラムは少なくとも１つのライセンスモデルを含み、

前記ソフトウェア実行プログラム（または前記ソフトウェア実行プログラムのうちの少なくとも１つ）は、前記トラステッド・モジュールに対して、前記ソフトウェア実行プログラムのデータのそれぞれが使用されることを要求するように動作可能であり、

前記要求に応答して、前記トラステッド・モジュール内の前記セキュア実行プログラムは、前記ライセンスモデル、または前記ライセンスモデルのうちの１つを用いてライセンス検査を実施し、ライセンス検査が成功すると、前記データをインストールすることを前記オペレーティング・システムへ要求するように動作可能である、

請求項５に従属する請求項６、または、これに従属する請求項７～２２のうちのいずれか１項に記載のコンピュータ・プラットフォーム。

【請求項２４】

前記オペレーティング・システムは、前記トラステッド・モジュールにのみ応答して、前記特定のデータを使用するようにプログラムされている、請求項２３のコンピュータ・プラットフォーム。

【請求項２５】

前記セキュア実行プログラムからオペレーティング・システムへのデータを使用するための要求は、前記専用通信路を介して供給される、請求項７に直接的または間接的に従属する請求項２０～２４のコンピュータ・プラットフォーム。

【請求項２６】

前記トラステッド・モジュールは、前記オペレーティング・システムへの前記データを使用するための要求を、ログ記録するように動作可能である、請求項１

９～２５のうちのいずれか１項に記載のコンピュータ・プラットフォーム。

【請求項２７】

ユーザ識別を含むさらなる着脱可能なトラステッド・モジュールをさらに含み、

前記第１のトラステッド・モジュールと前記着脱可能なトラステッド・モジュールとの間で認証検査を実施するように動作可能であり、

ライセンス検査時において、前記セキュア実行プログラムまたはソフトウェア実行プログラムは、前記ユーザ識別を参照して前記ライセンス検査を実施するように動作可能である、請求項１９～２６のうちのいずれか１項に記載のコンピュータ・プラットフォーム。

【請求項２８】

請求項３またはこれに従属する請求項４～２７のうちのいずれか１項に記載の第１のコンピュータ・プラットフォームから、請求項３またはこれに従属する請求項４～２７のうちのいずれか１項に記載の第２のコンピュータ・プラットフォームへ、データについてのライセンス（またはライセンスに関する鍵）を転送する方法であって、

前記トラステッドモジュール間でセキュアな通信を設定するステップと、

前記セキュアな通信を用いて前記第１のトラステッド・モジュールから前記第２のトラステッド・モジュールへ、前記ライセンスまたは前記鍵を送信するステップと、

前記ライセンスまたは前記鍵を前記第１のトラステッド・モジュールから削除するステップと、

からなる方法。

【発明の詳細な説明】

【０００１】

本発明は、コンピュータ・プラットフォームとその動作方法に関し、より詳細に述べれば、コンピュータ・プラットフォーム上のデータのインストール及び／又は使用を、制御及び／又は計測することに関する。

【０００２】

本明細書において、「データ」とは、画像、アプリケーション・ソフトウェア、ストリーミング・メディアなど、デジタル的に形成することが可能なものをいう。本明細書で説明される技術は、単純なテキスト文書から音声クリップおよび映像クリップ、ソフトウェア、グラフィック、光学素材およびマルチメディア素材まで、多様な種類の情報を保護または計測するために使用することができるであろう。

【０００３】

将来的には、コンピュータ・システムは、ウィルスまたは他の権限のない変更がオペレーティング・システムおよびインストールされたソフトウェアに施されていないことを保証するための他のコードに対する完全性検査を行うことにより、よりセキュアなブートが実現されるであろう。さらに、新世代の不正防止装置は市場にすでに登場している、あるいはまもなく登場することになるであろう。そしてこれらは、外部コンポーネントまたは可搬式コンポーネント（スマートカードなど）と内部コンポーネント（セキュリティ機能を有する埋込式プロセッサ、半埋込式プロセッサまたはコプロセッサ、すなわちマザーボード、ＵＳＢ、ＩＳＡ実装などを含むもの）との両方を含んでいる。これらの不正防止コンポーネントは、システムのハードウェアに不正が行われていないことを検査するために用いられ、現在利用可能なものよりも高い信頼度のマシン識別の形態（たとえば、そのマシンのイーサネット(R)名）を提供するであろう。しかしながら、いかにして不正行為を防止するか、さらにいかにしてソフトウェア開発者とエンドユーザとが許容可能な方法でソフトウェアをライセンスおよび計測するかは、依然、非常に重要な問題である。

【０００４】

ソフトウェア使用許諾は、ハッカーおよび不正行為にさらされており、現在使用されているすべてのソフトウェア・ライセンス方法は、これに関連する問題を有している。使用許諾のソフトウェアによる実施（「ライセンス管理システム」など）は柔軟性があるが、特にセキュアであるわけでも、高速であるわけでもない。特に、セキュリティが欠如していること（たとえば、一般的な「ハッキング」を受けているなど）、及び、ソフトウェアの正しい置き換えが難しいこと等の欠点がある。逆に、ハードウェアによる実施（「ドングル」）は、ソフトウェアによる実施よりも高速で一般的にセキュアであるが、柔軟性に欠けている。それらは、特定のソフトウェアにのみ対応して作られるものであり、エンドユーザにとっては不便なものである。

【０００５】

本発明は、その好適な実施形態において、セキュアで高速なハードウェアによる実施であるが、ソフトウェアによる実施の利便性及び柔軟性も有するという、両方の世界において最良のものを提供することを求めるものである。ソフトウェア使用許諾及び計測におけるこの新しい一般的な概念の実施例においては、不正防止ハードウェア内によりセキュアな鍵格納、暗号化機能及びよりセキュアな識別（すなわち認証）を伴う、コンピュータプラットフォームにおける完全性検査のセキュリティの改善が提供される。

【０００６】

先行特許出願（２０００年2月１５日付の国際特許出願第ＰＣＴ／ＧＢ００／００５２８号）は、完全性メトリックの信頼性のある計測および信頼性のある報告によって、コンピュータ・プラットフォームの完全性の検証を可能とするためのトラステッド・コンポーネントの使用について記載している。これは、ローカル・ユーザまたはリモートのエンティティのいずれによってもプラットフォームの完全性の検証を可能にする。この先行特許出願は、完全性メトリックを報告し、報告されたメトリックの値をメトリックの適切な値と比較することによって、プラットフォームの完全性の正確さを検証する一般的な方法について述べている。本発明は、先行特許出願の方法を用いて完全性が報告されるライセンス検査コードを使用する。この先行特許出願は、ここで参照することにより取り入れられ

るものとする。

【０００７】

概して、本発明の実施例は、コンピュータプラットフォームの不正（改ざん）防止コンポーネントまたは「トラステッド・モジュール」を、好ましくはその不正防止コンポーネントの内部で動作するソフトウェアと共に使用し、そのプラットフォームに対するデータのアップロード及び使用を、プラットフォームに対する一般的なドングルとして制御する。ライセンス検査は、信用できる環境（すなわち、ユーザが期待するようにふるまうことが信用できる環境）内で行われる。そしてこれは、アップロードおよびライセンス検査のソフトウェアの完全性検査により実施されることができる。計測記録が不正防止装置に格納され、必要に応じて管理者に報告されることができる。データに対する登録および支払いを可能とするための関連するクリアリングハウス機構が存在することが可能である。

【０００８】

より正式に述べれば、本発明の第１の態様によれば、内部改ざんに対して抵抗力があり、第三者の公開鍵証明書を格納するトラステッド・モジュール；プラットフォームまたはプラットフォームのユーザが特定のデータの使用を許諾されているか否かを検査し、そのデータを使用するため、及び／又は、そのデータの使用の監視をするためのインタフェースを提供するための（好ましくは一般的な）セキュア実行プログラムと、プラットフォームまたはプラットフォームのユーザが特定のデータのインストールを許諾（ライセンス）されているか否かを検査し、及び／又は、インストール前にデータの完全性を検査する（好ましくは一般的な）セキュア・ローダとのうちの少なくとも１つを含むライセンス関連コードを格納するための手段；第三者の秘密鍵を用いて署名されたライセンス関連コードのハッシュされたバージョンを格納する手段、これらが設けられ、ブート時に、ライセンス関連コードが、前記署名されたバージョン及び前記公開鍵証明書を参照して完全性の検査が行われ、この完全性検査が失敗した場合、ライセンス関連コードはロードされないようにプログラムされたコンピュータ・プラットフォームである。

【０００９】

本明細書のコンテクストにおいて、「ユーザ」という用語は、プラットフォームのエンドユーザ、またはシステム・管理者またはその両方を意味することが可能である。

【００１０】

上述の先行特許出願において説明されるように、トラステッド・モジュールまたはコンポーネントは、内部データの権限のない変更または閲覧に対して影響をうけないことが好ましい。偽造を防止することは物理的なことであり、偽造を防止することが不正防止であり、偽造を防止するためには、長距離において、セキュアに通信するための暗号化機能を有することが好ましい。トラステッド・モジュールを構築する方法は、本来、当業者にとって周知のものである。トラステッド・モジュールは、それ自体に暗号化識別子を付与し、信憑性、完全性、機密性、再生攻撃に対しての保護を提供し、デジタル署名を作成し、および必要に応じてデジタル証明書を使用するための暗号化手段を使用することができる。これらの及びその他の暗号化手段とそれらの初期化は、セキュリティの当業者には周知のものである。

【００１１】

この完全性検査は、ライセンス関連コードを読み込みハッシュして第１のハッシュを生成し、この署名されたバージョンを読み込み公開鍵証明書を用いて復号化して第２のハッシュを生成し、前記第１及び第２のハッシュを比較することにより行われることが好ましい。

【００１２】

ライセンス関連コードは、トラステッド・モジュールと別のコンピュータ・プラットフォームのさらなるトラステッド・モジュールとの間でライセンス鍵が転送されることを可能にするセキュア鍵転送コードも含むことが好ましい。この鍵転送コードは、ロック解除鍵を含むライセンス・モデルを使用している場合であって、データが暗号化された形態で送信され、この保護されたデータを復号して実行可能にするためにロック解除鍵が用いられるようなライセンス・モデルである場合に、鍵管理を改善するという点で特に有効である。この転送は、公開鍵インフラストラクチャを用いてロック解除鍵を含むメッセージを暗号化し、ハッシ

ュ化及びデジタル署名によって完全性を検査することによって実行することができる。このセキュア・ローダ用いて、データ自体を転送するということも可能である。

【００１３】

ライセンス関連コードは、トラステッド・モジュールと通信するために呼び出されることが可能なインタフェース・サブルーチンのライブラリも含むことが好ましい。このクライアント・ライブラリは、アプリケーションがトラステッド・モジュールと通信するために呼び出す上位インタフェース・サブルーチンの集合である。また、このクライアント・ライブラリは、トラステッド・モジュール及びオペレーティング・システム（「ＯＳ」）との通信のため、ソフトウェア実行プログラム（後述）により利用されることが可能である。

【００１４】

ライセンス関連コードは、少なくとも１つのデータのグループに対し、個々のデータのグループを指定しそのデータのグループへのインタフェースとして機能するように動作可能な１つ（または複数）のソフトウェア実行プログラムを含むことが好ましい。これによって、保護されるデータに特化したライセンス保護の手段が可能になり、したがって、より高いレベルの保護が可能になる。ソフトウェア実行プログラムは、アプリケーションと関連づけられている場合、任意であるが、そのアプリケーションによる要求（ＡＰＩコール）を処理する。

【００１５】

空間的に許されるならば、トラステッド・モジュールによって少なくとも一部に、ライセンス関連コードを格納する手段、及び／又は、ライセンス関連コードのハッシュされたバージョンを格納する手段が設けられることが好ましい。

【００１６】

プラットフォームのトラステッド・モジュール及びオペレーティング・システムは、そのコンピュータ・プラットフォームの他の部分へはアクセスできない専用通信路を有することが好ましい。

【００１７】

次に、これらのコンポーネントにおいて、汎用データ使用許諾のためのシステ

ムを形成するために対話する方法が考察される。このようなシステムは、いくつかのステージにより構築され、これらは相互に進行していると考えられる。第１のステージでは、ドングルなどの現在の使用許諾方法を改善し、一般的なドングルとして機能するトラステッド・モジュールを作成する。このトラステッド・モジュールは、一般のライセンス関連ソフトウェアの制御下におかれ、使用許諾検査を行い、また、完全性検査を行うことによりバイパスされることを防止している。このようなライセンス検査ソフトウェアは、トラステッド・モジュールそれ自体の内部で動作する必要はない。好適なステージは、使用許諾ソフトウェアがトラステッド・モジュール内で動作するシステムの論理的な拡張である。あるデータのロードまたは実行の要求は、好ましくは、ソフトウェア実行プログラムからトラステッド・モジュールへ送信される。トラステッド・モジュール内の使用許諾ソフトウェアは、ライセンスの詳細に基づいてこの要求を評価し、許可するか否かを判定する。要求が許可される場合、この情報は、トラステッド・モジュールからＣＰＵへのハードウェア通信路を介してＯＳへ伝送される。この通信路は、通常のアプリケーションとＯＳ以外のソフトウェアへはアクセス不可であることが好ましい。その後、ＯＳは、データをロードまたは実行するための処理を適切に開始する。

【００１８】

ここで、システム・コンポーネントが便利な使用許諾機能を実行するために対話可能なさまざまな方法が考えられる。第１の考えとしては、セキュア・ローダがデータをインストールするように動作する方法が考えられる。

【００１９】

ーインストール態様において、オペレーティング・システムは、プラットフォームまたはそのユーザ（エンドユーザまたはシステム管理者）がその特定のデータをインストールし、及び／または、そのデータの完全性を検査することが許可されているか否かのライセンス検査を行うことをセキュア・ローダへ要求するように動作可能であり、この応答にしたがって、前記オペレーティングシステムは、その特定のデータをインストールする、または、インストールしないように動作することが可能である。プラットフォームまたはユーザに対するこの検査は、

トラステッド・モジュール内またはスマートカード内の秘密アプリケーション鍵または他の秘密の存在についての検査、または、トラステッド・モジュールまたはスマートカードの識別及び存在の検査など、様々な方法により実施可能である。このような識別は、開発者に知らされることが可能であり、または、そのような秘密は、登録処理中にトラステッド・モジュールまたはスマートカードに挿入されることも可能である。この処理は、後述する例Ａのプロセスに類似している。

【００２０】

この態様において、オペレーティング・システムは、セキュア・ローダのみに応答して、特定のデータをインストールするようにプログラムされていることが好ましい。：また、この態様では、トラステッド・モジュールは、インストールされる特定のデータと関連する関係者についての公開鍵証明書を格納し；オペレーティング・システムは、検査の要求において、特定のデータを、そのデータを関係者の秘密鍵で署名したハッシュしたバージョンと一緒に含めるように動作可能であり；検査の実施において、セキュア・ローダは、要求に含まれる特定のデータをハッシュして第３のハッシュを生成し；要求に含まれる署名されてハッシュされたバージョンを、前記関係者の公開鍵証明書を用いて復号して第４のハッシュを生成し；前記第３及び第４のハッシュが適合するか否かに従って応答を生成するように動作可能であることが好ましい。

【００２１】

これは、メッセージの完全性を検査するものである。また、この完全性検査機構は、チャレンジ／レスポンスまたは前記ハッシュに通信履歴を導入するなどの標準的な機構を利用することにより再生攻撃も防止する。非否認の問題は、不正防止ハードウェアに秘密鍵を保持することにより回避されることができる。この検査のための要求は、特定のデータについてのソフトウェア実行プログラムを含むことが好ましい。

【００２２】

別のインストール態様において、ソフトウェア実行プログラム（またはソフトウェア実行プログラムのうちの少なくとも１つ）は、特定のデータをインストー

ルすることをトラステッド・モジュールへ要求するように動作可能であり；そのような要求に応答して、トラステッド・モジュール内のセキュア・ローダは、プラットフォームまたはそのユーザが、その特定のデータをインストールし、及び／または、そのデータの完全性を検査することを許可されているか否かをライセンス検査し、この検査の結果をオペレーティングシステムへ応答するように動作することが可能であり；この応答に従って、オペレーティング・システムは、特定のデータをインストールする、またはインストールしないように動作可能である。

【００２３】

この検査は、上記の一インストール態様に関連して説明されたものと同様の方法で実行されることが可能である。

【００２４】

この別の態様においては、オペレーティングシステムはトラステッド・モジュールにのみ応答して特定のデータをインストールするようプログラムされることが好ましい。また、この態様においては、上述のように、トラステッド・モジュールからオペレーティング・システムへの応答は、専用通信路を介して供給されることが好ましい。

【００２５】

いずれのインストール態様においても、検査が成功した場合、トラステッド・モジュールは、特定のデータを監査するためにログを生成するように動作可能であることが好ましい。また、検査が成功した場合、セキュア・ローダは、その特定のデータに対してウィルス検査を実施するように動作可能であることが好ましい。

【００２６】

インストール時において、前記特定のデータはトラステッド・プラットフォームにインストールされる。代替として、プラットフォームは、さらなる着脱可能なトラステッド・モジュール（スマート・カードなど）を含み、第１のトラステッド・モジュールと着脱可能なトラステッド・モジュールとの間で認証検査を実施するように動作することも可能であり、この場合、インストール時には、特定

のデータはこのさらなるトラステッド・モジュール内にインストールされる。

【００２７】

ソフトウェア実行プログラムは、それ自体、セキュア・ローダにより実行される完全性検査により保護されることが可能である。たとえば、この処理は以下のように行われる。

（ａ）ソフトウェア実行プログラムは、クライアントのトラステッド・モジュールに対応する公開鍵を含むようにカスタマイズされる。；

（ｂ）カスタマイズされたソフトウェア実行プログラムに関連するデータがクライアントへ送信される。；

（ｃ）データ及びソフトウェア実行プログラムの両方が、クリアリングハウス／開発者の秘密鍵を用いてハッシュおよび署名され、これがデータおよびソフトウェア実行プログラムと共に送信される。；

（ｄ）これを受信すると、セキュア・ローダがソフトウェア実行プログラムの完全性を検査する－ソフトウェア実行プログラムのインストール時、このパッケージは、ハッシュされ、（トラステッド・モジュールの公開鍵を用いて）復号された署名との比較により検証される。デジタル署名が期待する物に適合しない場合、ソフトウェア実行プログラムはロードされず、この場合セキュア・ローダはエラーを示す。また、セキュア・ローダは、同じ方法を用いてデータ自身の完全性の検査も行う。

【００２８】

ここで、セキュア実行プログラムがデータを使用するために動作する方法について説明する。

【００２９】

第１の実施態様において、ソフトウェア実行プログラム（またはソフトウェア実行プログラムのうち少なくとも１つ）は、トラステッド・モジュールの公開鍵とそれぞれのデータについてのライセンスモデルとを含み；オペレーティング・システムは、そのそれぞれのデータが使用されることを、ソフトウェア実行プログラムへ要求するように動作可能であり；そのような要求に応答して、ソフトウェア実行プログラムは、そのライセンスモデルを使用して、プラットフォームま

たはそのユーザがそのデータの使用を許諾されているか否かをライセンス検査するように、セキュア実行プログラムへ要求するように動作可能であり；そのような後者の要求に応答して、セキュア実行プログラムは、要求されたライセンス検査を実行し、トラステッド・モジュールの秘密鍵を用いてこのライセンス検査の結果に署名し、署名された結果をソフトウェア実行プログラムへ応答するように動作可能であり；そのような応答に応答して、ソフトウェア実行プログラムは、トラステッド・モジュールの公開鍵を使用して署名された結果の完全性を検査し；ライセンス検査が成功した結果として完全性検査が成功すると、そのデータを使用するようにオペレーティング・システムへ要求するように動作可能である。

【００３０】

第２の実施態様において、ソフトウェア実行プログラム（またはソフトウェア実行プログラムのうち少なくとも１つ）は、トラステッド・モジュールの公開鍵とそれぞれのデータについてのライセンスモデルとを含み；オペレーティング・システムは、その特定のデータが使用されることを、セキュア実行プログラムに要求するように動作可能であり；そのような要求に応答して、セキュア実行プログラムは、特定のデータに関するライセンスモデルについて、トラステッド・モジュールの秘密鍵を使用して署名された要求を、それぞれのソフトウェア実行プログラムへ送信するように動作可能であり；そのような後者の要求に応答して、そのソフトウェア実行プログラムは、トラステッド・モジュールの公開鍵を使用して要求の完全性を検査し、完全性検査が成功すると、セキュア実行プログラムへライセンスモデルを送信するように動作可能であり；ライセンスモデルの受信時に、セキュア実行プログラムは、そのライセンスモデルを使用してライセンス検査を実行し；ライセンス検査が成功すると、そのデータを使用することをオペレーティング・システムへ要求するように動作可能である。

【００３１】

第３の実施態様において、セキュア実行プログラムは、少なくとも１つのライセンスモデルを含み；オペレーティング・システムは、その特定のデータが使用されることを、セキュア実行プログラムへ要求するように動作可能であり；そのような要求に応答して、セキュア実行プログラムは、ライセンスモデルまたはラ

イセンスモデルのうち１つを使用してライセンス検査を実施し；ライセンス検査が成功すると、そのデータを使用することをオペレーティング・システムへ要求するように動作可能である。

【００３２】

これらの３つの実施態様のいずれにおいても、オペレーティング・システムは、セキュア実行プログラムまたはソフトウェア実行プログラムのみに応答して、特定のデータを用いるようにプログラムされることが好ましい。

【００３３】

第４の態様において、セキュア実行プログラムは少なくとも１つのライセンスモデルを含み；ソフトウェア実行プログラム（またはソフトウェア実行プログラムのうち少なくとも１つ）は、そのそれぞれのデータが使用されることをトラステッド・モジュールへ要求するように動作可能であり；そのような要求に応答して、トラステッド・モジュール内のセキュア実行プログラムは、ライセンスモデルまたはライセンスモデルのうち１つを使用してライセンス検査を実施し、ライセンス検査が成功すると、そのデータを使用することをオペレーティング・システムへ要求するように動作可能である。この場合、オペレーティング・システムは、トラステッド・モジュールにのみ応答して、特定のデータをインストールするようにプログラムされていることが好ましい。

【００３４】

第２～第４の実施態様のいずれにおいても、データを使用するためのセキュア実行プログラムからオペレーティング・システムへの要求は、専用通信路を介して供給されることが好ましい。

【００３５】

第１から第４の実施態様のいずれにおいても、トラステッド・モジュールは、データを使用するためのオペレーティング・システムへの要求を、ログするように動作可能であることが好ましい。使用許諾または計測のセキュリティおよび信頼性は、トラステッド・モジュール内でのデータ使用をセキュアにログ記録することにより向上される。使用許諾関係のアクティビティのログ記録が実行され、不正防止コンポーネント内にセキュアに記録される。使用許諾を行う間、これを

実行するための多数の様々なオプションが存在する。もっとも一般的なのは、データがセキュア実行プログラムまたはソフトウェア実行プログラムによって、実行可能となるステージである。別の一般的な時点は、セキュア・ローダがインストールされるデータに対する完全性検査を良好に完了し、このデータをクライアント・マシンへ良好にインストールし終えたときである。このセキュア実行プログラム、ソフトウェア実行プログラム、およびセキュア・ローダは完全性検査により保護されているため、ログ処理をバイパスまたはログを編集しようとするハッカーに対して何らかの保護が与えられる。そのようなログは、セキュアな監査情報と、使用毎の支払い、レンタル、時間依存課金などの柔軟な使用許諾および支払いモデルの可能性との両方を提供する。このような監査ログは、マシン・ユーザのＩＴ部門または社内監査担当者などの第三者がアクセス可能な使用報告および情報に対する基盤を形成する。また、それらは、宣伝活動、または評価額に対するフィードバックなど、商業上の価値も有する。

【００３６】

さらに、プラットフォームが、上述のようなさらなる着脱可能なトラステッド・モジュール（スマートカードなど）を含む場合、プラットフォームがユーザ識別を含み、ライセンス検査時、セキュア実行プログラムまたはソフトウェア実行プログラムが、このユーザ識別を参照してライセンス検査を実施するように動作可能することが好ましい。

【００３７】

ユーザが、ソフトウェアの実行または保護されたデータへのアクセスを要求する場合、セキュア実行プログラムは、たとえば次のようにライセンス検査を実施できる。

（ａ）装置において、ソフトウェアまたはデータ参照に対応する秘密を検査する。または

（ｂ）ロック解除鍵を用いてデータを復号し、データを実行可能にする（コードの部分的なロック解除など、ロック解除鍵の機能を差別化するさまざまなオプションが存在する）。または

（ｃ）データ参照および装置識別に対応するデータベース内の使用許諾の権利を

検査する。または

（ｄ）データベースからデータ参照および装置識別に対応する鍵を取得し、これを用いてデータをロック解除する。

【００３８】

ユーザがアプリケーションを動作させようとする場合、セキュア実行プログラムが全体的な制御を引き受け、もしデータに関連する情報が存在するならば、これをソフトウェア実行プログラムから取得し、いずれの種類の検査が開発者によって指定されたかを明らかにする、というように準備される。検査の種類が特定されると、セキュア実行プログラムがこれを実行し、特定されない場合は、後述するようにデフォルトの検査が用いられる。この検査が成功した場合、セキュア実行プログラムはデータを実行する。検査が失敗した場合、セキュア実行プログラムはデータが実行されることを防止する。

【００３９】

ソフトウェア実行プログラムがライセンス方法を指定しない場合、またはアプリケーションに添付されたソフトウェア実行プログラムが存在しない場合、セキュア実行プログラムは、特定のマシンのために定義されたデフォルト・プロトコルを使用することができる。これは、マシンの環境を考慮してマシンの管理者により設定される。たとえば、マシンがただ１人にしか使用されない場合、内部トラステッド・モジュールに対応するライセンスモデルが、おそらく最も適切であろう。このセキュア実行コードは、ブート時の完全性処理の一部として、プラットフォームの完全性検査に含まれてしまっているため、このセキュア実行プログラムを、したがってライセンス検査を、バイパスすることは不可能である。

【００４０】

様々な使用許諾のモデルが、さまざまな方法によりセキュア実行プログラム及びソフトウェア実行プログラムを利用する。上記からも明らかなように、それらを組み合わせて使用することも可能であり、または、その一方をライセンス検査の実施に用いることも可能である。主たる好ましいオプションは次の２つである。：

（１）第１のオプションは、それぞれのデータの一部に添付される様々ソフトウ

ェア実行プログラムを有し、これら特定のデータの一部について、セキュア実行プログラム内でライセンス検査を支配することである。次のセクションの例のうちいくつかにおいて、ソフトウェア実行プログラムは、このようにしてオペレーティング・システムと直接通信する。

（２）代替のアプローチは、検査を実行するプラットフォーム内に汎用コードを構築し、ＯＳと何らかのソフトウェア実行プログラムとの間のブリッジとしてセキュア実行プログラムを動作させることにより、セキュア実行プログラムをもっと重要視することである。この代替案は、開発者に対してプロトコル記述の負荷をかけないようにし、開発者が非常に容易に使用許諾の選択を指定できるようにし、プラットフォームの完全性検査が行われた場合にライセンス検査コードの完全性検査を利用する。

【００４１】

データの一部に関連付けられたソフトウェア実行プログラムは、（登録処理の間に得られた）検査されるべき特定の情報と共に、その使用許諾に用いられる方法についてコンピュータプラットフォーム内のセキュア実行プログラムへ知らせる情報も有する。検査が行われる特定のトラステッド・デバイス及びそのデータへの参照は、保護される。例えば、たとえば、ｌｉｃｅｎｓｉｎｇ＿ｍｅｔｈｏｄ（秘密，ｓｃ，ｋ，ｗ）とｌｉｃｅｎｓｉｎｇ＿ｍｅｔｈｏｄ（秘密，ｔｃ，ｋ，ｗ）は、ｗで参照されるソフトウェアを示しており、秘密ｋが、マシンの現在のスマートカードまたは内部トラステッド・コンポーネント内に格納されていることがわかった場合のみ、そのマシン上で実行することが許可されなければならないことを示している。

【００４２】

さまざまなソフトウェア実行プログラムがデータに対して添付され、ソフトウェア実行プログラムは、いずれの種類のライセンスモデルが使用されるかを示す。セキュア実行プログラムは、このライセンスモデルに従って動作時（ランタイム）に検査を実行し、検査が成功しない限りソフトウェアｗを実行することを許可しない。これらの手段により、クリアリングハウスからトラステッド・モジュールへの通信は、クリアリングハウスがいずれの使用許諾プロトコルを使用する

ことを望んでいるかを特定する。

【００４３】

様々な種類のプロトコルがセキュア実行プログラムにより採用されてよい。たとえば、第１のプロトコルとしては：

・セキュア実行プログラムは、トラステッド・モジュールＩＤエントリまたはスマートカードＩＤエントリを検査する。；

・任意で、セキュア実行プログラムは、データベース・エントリを、トラステッド・モジュール内に格納されたプロファイル内へダウンロードする。；

・セキュア実行プログラムは、データのロック解除鍵について、データ参照およびトラステッド・モジュールＩＤエントリ（またはスマートカードＩＤエントリ）に対して、外部データベース、またはトラステッド・モジュール内に格納されたプロファイルを検査する。；

・セキュア実行プログラムは、この鍵を取得し、関連づけられたデータを復号して、オペレーティング・システムがこれを実行できるようにする。；

・任意で、セキュア実行プログラムは、データ参照と共にロック解除鍵をトラステッド・モジュール内に格納する。；

・データは、対応する鍵を用いて暗号化または部分暗号化により保護される。；

・このロック解除鍵の機能を異ならせるさまざまなオプションが存在する。；

・支払いと引き換えに、該トラステッド・モジュールＩＤに対応するデータベース・エントリは、この鍵を用いて更新される。

【００４４】

第２のプロトコルとしては：

・任意で、セキュア実行プログラムは、データベース・エントリをトラステッド・モジュール内に格納されたプロファイル内へダウンロードする。；

・セキュア実行プログラムは、ライセンスについて、データ参照およびトラステッド・モジュールＩＤエントリ（またはスマートカードＩＤエントリ）に応じて、外部データベース、またはトラステッド・モジュール内に格納されたプロファイル内を検査する。；

・適切なライセンスが存在した場合にのみ、セキュア実行プログラムはＯＳに対

してそのデータを実行する権限を与える。；

・支払いと引き換えに、該トラステッド・モジュールＩＤまたはスマートカードＩＤに対応するデータベース・エントリは、適切な許可に更新される。

【００４５】

第３のプロトコルとしては：

・セキュア実行プログラムは、ソフトウェアまたはデータベース参照に対応する秘密について、トラステッド・モジュール（スマートカードを含む）を検査する。；

・検査される秘密は、ライセンスが検査済みであるデータに関連するソフトウェア実行プログラムによって指定される。；

・秘密がトラステッド・モジュールに存在する場合のみ、セキュア実行プログラムはＯＳに対して関連するソフトウェアまたはデータを実行する権限を与える。

【００４６】

第４のプロトコルとしては：

・セキュア実行プログラムは、トラステッド・モジュールまたはスマートカード内に格納された何らかのデータに関連するロック解除鍵を用いてデータを復号し、オペレーティング・システムがこれを実行できるようにする。；

・コードの部分的なロック解除を含めて、ロック解除鍵の機能を異ならせるさまざまなオプションが存在する。

【００４７】

第５のプロトコルとしては：

・セキュア実行プログラムは、トラステッド・モジュールまたはスマートカード内に格納された何らかのデータに関連する鍵、または、キーボードを介してユーザから入力された他のもの、トラステッド・モジュールＩＤまたはスマートカードＩＤ、及び、所定のアルゴリズムを用いて、復号化鍵を計算する。；

・セキュア実行プログラムは、復号化鍵を用いてデータを復号し、オペレーティング・システムがこれを実行できるようにする。；

・コードの部分的なロック解除を含めて、復号化鍵の機能を異ならせるさまざまなオプションが存在する。

【００４８】

第６のプロトコルとしては：

・セキュア実行プログラムは、１グループのユーザに対し浮動ライセンスの使用を許可する。；

・セキュア実行プログラムは、そのデータについてのライセンス鍵を得るため、トラステッド・モジュールＩＤエントリまたはスマートカードＩＤエントリに対して、データベースを検査する。；

・セキュア実行プログラムは、その特定の実行を可能にするために、（もし利用可能であれば）ライセンス鍵を取得する。；

・セキュア実行プログラムは、データの実行が終了するとライセンス鍵をプールへ返す。

【００４９】

第７のプロトコルとしては：

・セキュア実行プログラムは、第１～第６のプロトコルの組み合わせを実施し、様々なデータ・エンティティについて、様々なライセンス検査の方法が用いられるようにする。；

・プロトコルの選択は、セキュア実行プログラム自体により行われてよい。；

・デフォルトまたは上書きのプロトコルは、管理者により定義されることが可能である。；

・特定のデータについてライセンスの検査時に使用されるプロトコルは、そのデータに関連する任意のソフトウェア実行プログラムにより判定される。

【００５０】

本明細書で後述されるライセンスモデルには、データの複製を防止しないものもあり、権限のないデータの利用を禁止し、プラットフォームの一部として不正防止装置を有するマシンの利用のログ取得を確実にするにすぎない。データ保護に求められるレベルは、ビジネス・モデルに依存している。データは、従来の通信路および他の非セキュアな通信路を介して送信されることもできる。しかしながら、ライセンス鍵の転送がセキュアであることが最も重要とされる。

【００５１】

本発明の第２の態様によれば、各々が本発明の第１の態様に従う第１のコンピュータ・プラットフォームから第２のコンピュータ・プラットフォームへライセンス（またはそのための鍵）を転送する方法であって、トラステッド・モジュール間にセキュアな通信を設けるステップと、第２のセキュアな通信を用いて第１のトラステッド・モジュールから第２のトラステッド・モジュールへライセンスまたはライセンス用の鍵を送信するステップと、前記ライセンスまたはライセンス用の鍵を第１のトラステッド・モジュールから削除するステップとから成る方法が提供される。

【００５２】

顧客が、ライセンスを他の人または他のマシンへ転送したいと考える多くの状況が存在する。たとえば、新しいＰＣが購入された場合、ソフトウェアがアップグレードまたは入れ替えされる場合、または顧客がデスクトップ機のかわりに携帯機でアプリケーションを動作させたい場合などがある。それぞれのアプリケーションに特化されたハードウェア・ドングルを移動することは容易な解決策であり、ある種のスマートカードを利用する類似の解決策も存在する。汎用ドングルを提供するすべてのシステム、これらはユーザにとってほとんどの状況においてより実用的であるが、現状において鍵管理の重大な問題に直面している。Ｗａｖｅ　Ｓｙｓｔｅｍ社の「ＷａｖｅＮｅｔ」とライセンス管理システム（「ＬＭＦｓ」）も例外ではない。ソフトウェアのみによる方法は、インストール／アンインストール処理を必要とし、加えて、同一のライセンスに対して第２のパスワードが発行される場合、エンドユーザが正規に購入したライセンスの数のみを使用していることを信用しなければならない。

【００５３】

トラステッド・モジュールを用いるライセンス移転のためのオプションは、選択されるライセンスの態様に依存する。一般的に、これらは次のようなものである。

【００５４】

データベース検査を用いる使用許諾を行うためには、両方のマシンのトラステッド・モジュールＩＤ（ライセンスが他のマシンへ移行される場合）、または、

両方のスマートカードＩＤ（ライセンスが他の人へ移行される場合）に対応するデータベース・エントリが、変更されなければならない。

【００５５】

トラステッド・モジュールに関連する指紋検査を伴う使用許諾、または、トラステッド・モジュール用にあつらえられたコードを使用する使用許諾を行うためには、新しい装置（すなわち他の人へライセンスが移行する場合はスマートカード、他のマシンへライセンスが移行する場合は内部トラステッド・モジュール）がベンダに再登録され、この新しい装置ＩＤに基づいて発行される別の鍵またはあつらえられたソフトウェアが、それぞれ取得されなければならない。

【００５６】

暗号化およびロック解除鍵を伴う方法において、アプリケーション毎に１つのスマートカードが存在する場合、適切なスマートカード（および識別番号）が、新しく許可される者に対して与えられなければならない。もしくは、ロック解除鍵およびデータは、ベンダが転送の報告を受け取る以上にかかわる必要なく、トラステッド・モジュール間で自動的に転送されることが可能である（第８の方法において、説明する）。この方法は、関連するデータの完全性検査、あるトラステッド・モジュールから別のトラステッド・モジュールへのライセンス鍵の複製、および、元のトラステッド・モジュールからのライセンスのアンインストールを含む。

【００５７】

クライアントマシンＭ１のＴＣ１からマシンＭ２のＴＣ２へデータＳについてのライセンス（すなわちロック解除鍵Ｌ）を転送するステージは、たとえば次のようになる。

【００５８】

Ａ．セキュア鍵転送コード（「ＳＫＴ」）は、ＢＩＳ処理の拡張として完全性チェックが行われる。このライセンス転送コードは、製造業者の秘密鍵を用いてハッシュおよび署名される。プラットフォームのブート／インストールの際、このパッケージは、製造業社によってトラステッド・モジュールにインストールされた公開鍵証明書を用いて、ハッシュ化及び復号された署名との比較により検証

され、完全性検査が行われる。デジタル署名が期待するものに適合しなかった場合、ライセンス転送コードはロードされず、プラットフォームの完全性検査は失敗する。

【００５９】

Ｂ．初期化。コンテンツ・プロバイダは、元の登録およびデータ・インストール処理によって、ＴＣ１の公開鍵をすでに保持している。もし、保持していない場合には、これがコンテンツ・プロバイダへ送信される。

１．ＴＣ１の所有者がＴＣ２へライセンスを転送したい場合、マシンＭ１のＯＳからＭ１内のＳＫＴへ、データＳのライセンスをＴＣ２へ転送するための呼び出しがある。

２．Ｍ１のＳＫＴは乱数Ｒを生成し、ライセンスの転送を要求しているＭ２へメッセージを送信する。このメッセージには、データＳへの参照とともにＴＣ１の公開鍵証明書が含まれる。

３．Ｍ２が適切な発信元からの認証を得ると、Ｍ２のＳＫＴは、Ｒ、ＴＣ２の公開鍵証明書、Ｓへの参照、および生成された新しいノンス(nonce)Ｔを含む肯定応答を返す。

４．その後、Ｍ１のＳＫＴは、ＴとともにＳのコンテンツ・プロバイダの公開鍵証明書をＭ２へ送信する。；

これらの通信は、送信者のマシン内のトラステッド・モジュールの秘密鍵により署名された通信のハッシュされたバージョンへ付加され、受信者のＳＫＴがメッセージの完全性を検査できるようにしている。この完全性検査が失敗した場合、メッセージは、それぞれのＳＫＴによりこれらのマシン内のＯＳへ送信され、プロトコルは停止する。

【００６０】

Ｃ．プログラム・アップロード。上記の認証が成功すると、ＴＣ１は、データＳ（任意でコンテンツ・プロバイダによって、すでに署名されたバージョン）をハッシュし、ＴＣ１の秘密鍵を用いて（たとえば、マイクロソフト社のＡｕｔｈｅｎｔｉｃｏｄｅを使用して）これに署名する。その後、ＴＣ１は、この署名をデータとともにＴＣ２内へアップロードする。任意で、このデータは暗号化され

る。

【００６１】

Ｄ．コード検証。ＴＣ２内のセキュア・ローダは、データＳの署名を次のように検証する：まず、ＴＣ１の公開鍵を使用してその署名を検査し、それによって、メッセージ・ハッシュを取得する；次に、Ｓのハッシュを計算し、復号されたメッセージ・ハッシュと同じであることを検査する。この検証が成功すると、セキュア・ローダは、ＴＣ２に対応するマシン内へプログラムをインストールする。成功しなかった場合は、ライセンス転送プロトコルのさらなる進行を阻止するエラー・メッセージをＳＫＴへ生成する。

【００６２】

Ｅ．転送鍵。Ｍ１のＳＫＴは、乱数発生器を使用して対称鍵を生成し、これを用いてロック解除鍵を転送するメッセージを暗号化する。Ｍ１のＳＫＴは、このメッセージを、ＴＣ２の公開鍵を用いて暗号化された対称鍵、及び、ＴＣ１の秘密鍵を用いて署名されたこの情報すべてのハッシュとともに、Ｍ２のＳＫＴへ送信する。ＴＣ２のみが、ロック解除鍵の復号を可能にする対称鍵を復号するためのＲＳＡ秘密鍵を有することになる。

【００６３】

Ｆ．メッセージ検証。Ｍ２のＳＫＴは、ＴＣ１の公開鍵を使用して署名を検査し、ＴＣ２の秘密鍵を用いて復号することにより得られた対称鍵を用いてメッセージを復号し、それによって、ロック解除鍵を取得する。署名が正しければ、この鍵は、トラステッド・コンポーネント内に格納され、データＳと関連づけられる。署名が正しくなければ、Ｍ１のＳＫＴへエラー・メッセージが送信され、プロトコルは停止する。

【００６４】

Ｇ．ＴＣ１からの鍵削除とコンテンツ・プロバイダ通知。Ｍ１のＳＫＴは、ロック解除鍵をＴＣ１から削除し、このログをＴＣ１に記録する。Ｍ１のＳＫＴは、ＴＣ１の秘密鍵を使用して署名され、コードＳのライセンスがＭ２へ移行されたことを通知するメッセージを、コンテンツ・プロバイダへ送信する。任意で、Ｍ１またはＭ２のＳＫＴは、Ｍ２の所有者がどのように登録のための交信を持て

るかの詳細を与えるメッセージを、データ・ベンダへ送信する。

【００６５】

トラステッド・コンポーネント及びソフトウェア実行プログラムには、特定の機能に対する環境を提供することによって、オペレーティング・システムの新しい部分として動作し、オペレーティング・システムとアプリケーションとの間のブリッジを形成するためのオプションがある。たとえば、「保存」および「リストア」などのＡＰＩコールが、トラステッド・モジュールに対して行われることが可能である。「保存」は、データをトラステッド・モジュールを通過させ、トラステッド・モジュールでデータを暗号化し、これをトラステッド・モジュールまたはハードディスクのいずれかに格納する。トラステッド・モジュールの許可なくこのデータにアクセスすることはできなくなる。そのようなデータを使用してトラステッド・モジュール内で何らかの変形を実行し、ソフトウェアがＡＰＩコールを使用して、トラステッド・モジュールから情報を要求し、出力された応答を取得するための、さらなるオプションが存在する。概して、ＡＰＩコールは、ソフトウェア実行プログラムまたはアプリケーション・コードから、トラステッド・モジュールにトラステッドモジュールまたはトラステッド・モジュール上に格納された秘密アプリケーション鍵の存在を検査させ、さらに、特定の機能またはデータ格納のための環境を提供するためにトラステッド・モジュールを用いるために利用されることができる。

【００６６】

具体的に述べると、ＡＰＩコールは、アプリケーション・コードまたはソフトウェア実行プログラムに追加され、クライアント・ライブラリを介してＯＳ、トラステッド・モジュール、またはセキュア実行プログラムへ問い合わせを行うために使用される。たとえば、ＡＰＩコールは、アプリケーション・コードまたはソフトウェア実行プログラムに追加され、クライアント・ライブラリを介してＯＳ、トラステッド・モジュール、またはセキュア実行プログラムへ問い合わせを行うために使用され、トラステッド・モジュールまたはスマートカード内の秘密アプリケーション鍵または他の秘密の存在を検査すること、または、トラステッド・モジュールまたはスマートカードの識別および存在を検査することに使用さ

れる。

【００６７】

後に詳述されるある特定のモデルでは、トラステッド・モジュールのＩＤに対応するライセンス関連データベースのエントリが更新されるライセンスモデルが採用され、いったんこのデータベース上の許可が検査されおわると、セキュア実行プログラムはデータの実行のみを許可する。この場合、アプリケーションと関連づけられたソフトウェア実行プログラムは、セキュア実行プログラム（おそらくトラステッド・モジュール内）を呼び出し、セキュア実行プログラムがライセンスを検査し、この検査が成功すると、アプリケーションが通常の方法で実行されるように、その呼び出し（コール）をオペレーティング・システム（ＯＳ）へ渡す。言いかえるならば、ＯＳは、その呼び出しが、セキュア実行プログラムまたはソフトウェア実行プログラムなどのセキュアなライセンス関連コードからの呼び出しである場合にのみ、データを実行するための呼び出しを受け入れる。

【００６８】

後に詳述される別の特定のモデルでは、このトラステッド・モジュールは、本発明を実現するために使用されるハードウェア及び／又はソフトウェアを格納することが好ましく、また、ＯＳは、呼び出しがトラステッド・モジュールからのものであるならばデータを実行するための呼び出しを受け入れる。特に、トラステッド・モジュールは、アプリケーションとＯＳとの間のブリッジとして動作することが好ましく、ＯＳは、トラステッド・モジュールからの要求を除いて、アプリケーションをロードするためのすべての要求を無視することが好ましい。

【００６９】

１つの可能なライセンスモデルでは、データのロック解除鍵を得るため、セキュア実行プログラムに、トラステッド・モジュールＩＤエントリに対して、データベースを検査させることである。この場合、データは、対応する鍵を使用して暗号化または部分暗号化により保護され、したがって、盗用される恐れなく自由に配布されることができる。支払いが行われると、トラステッド・モジュールのＩＤに対応するデータベース・エントリは、この鍵を用いて更新される。ユーザがアプリケーションを動作させたい場合、この鍵は、データのロック解除を許可

するために取得されることが可能である。その後、この鍵は、不正防止装置内に格納され、データベース検索が一回だけで済むようにしている。しかしながら、浮動ライセンスが所望されるライセンスモデルにおいては、そのような鍵は一元的に格納し、それぞれの実行時にのみアクセスを許可し、ライセンスが別のユーザによる使用のために適切なグループへ返還されるようにする方がより望ましい。このようにして、ライセンス「交換」に対するモデルが提供される。

【００７０】

したがって、本発明は、セキュア実行プログラムまたはソフトウェア実行プログラムが、ソフトウェアのライセンス鍵のライセンス鍵のためトラステッド・モジュールＩＤエントリに対してデータベース内を検査を行い、特定の実行を可能とするためにライセンス鍵を取り出し（使用可能な場合）、アプリケーションが終了されたときにそのライセンス鍵をプールに返すことにより、ユーザのグループに対して浮動ライセンスを用いるため、セキュア実行プログラムまたはソフトウェア実行プログラムとトラステッド・モジュールとの間に自由な対話がある場合に拡張される。

【００７１】

ホット・デスキングなどのより柔軟な状況に適合するために、さまざまなユーザが汎用端末を使用している場合、複数のトラステッド（信頼）装置の組み合わせが使用可能である。特に、固定式不正防止コンポーネントと可搬式不正防止コンポーネントとの組み合わせは、使用許諾において、大幅な柔軟性を与える。当然ながら、個人のユーザのスマート・カードは、コンピュータ内の内部不正防止装置と組み合わせて使用される。この種のライセンスモデルの場合、ソフトウェア実行装置またはセキュア実行装置は、特定のスマート・カードが存在する（またはスマートカードの選択されたグループのうちの１つが存在する）場合にのみデータを実行させる。

【００７２】

この内部トラステッド・モジュールは、トラステッド・マシン識別を含み、可搬式トラステッド・モジュール（この場合は、スマートカード）は、ユーザ専用の識別を含む（導入された生物測定装置を使用して認証されることが可能である

）。使用許諾を行うための多種多様な方法が、そのような状況に使用されることが可能で（１つの例を後述する）あり、これらは「好適な実施形態」の部分で提示されるオプションに類似する。実現される特定のモデルによれば、その違いは以下の通りである。

・セキュア実行プログラムまたはソフトウェア実行プログラムにより実行されるライセンス検査には、内部マシン識別ではなく、スマートカード識別が含まれる。したがって、たとえば、マシン識別でなくユーザ識別がプロファイルまたはディレクトリに対して検査される。スマートカード上に格納されるロック解除鍵の場合において、トラステッド・モジュール内のスマートカードＩＤの存在は、ロック解除鍵を要求する際に、セキュア実行プログラムに、（ａ）トラステッド・モジュールの公開鍵を使用してロック解除鍵を暗号化するスマートカードによって、暗号化形態のロック解除鍵をトラステッド・モジュールへ複製させる、または（ｂ）スマートカードからロック解除鍵を直接使用させる。

・内部トラステッド・モジュールとスマートカードとの間に認証が存在する。スマートカードとトラステッド・モジュールとの間の認証は、スマートカードが挿入されるステージで実行され、現在のスマートカードＩＤがトラステッド・モジュール内に一時的に格納され、トラステッド・モジュールＩＤが本明細書で説明されるライセンスモデルで使用されたのと同じ方法で、ライセンス検査のために使用される（後述の例Ａ、Ｂ、およびＦを参照）。スマートカードが取り除かれると、または（単一の署名がなされて）ユーザがログアウトすると、トラステッド・モジュール内のこの一時的なスマートカードＩＤ値は、ヌル値にリセットされる。

【００７３】

ユーザベースの使用許諾とマシンベースの使用許諾との両方は、同一のマシン内の異なるデータに対して使用されることが可能である。これは、（ａ）トラステッド・モジュール内のスマートカードＩＤ値がヌルでないならば、マシンＩＤではなくスマートカードＩＤに対して（さらにこれが失敗した場合はマシンＩＤに対して）ディレクトリ・エントリを検査する、または（ｂ）スマートカードがリーダに現在挿入されているならば、スマートカード内のロック解除鍵を検査す

ることによって成される。すなわち、トラステッド・モジュールへ複製されるか、または、それを直接使用するか、のうちいずかの要求が行なわれる。

【００７４】

したがって、本発明は、可搬式トラステッド・モジュールと関連づけられたユーザ識別に基づいてライセンス検査を実施するための、内部マシン・トラステッド・モジュールと可搬式トラステッド・モジュール（およびセキュア実行プログラムとソフトウェア実行プログラム）の任意の組み合わせが用いられる場合に拡張される。

【００７５】

以下で詳述される本発明の使用許諾システムは、以下の特徴を有する。

・コンピュータ・プラットフォームは、第三者Ｃに登録される。任意で、Ｃは、トラステッド・モジュールＩＤまたはスマートカードＩＤを与えられる。；

・トラステッド・モジュールとＣとの間の認証と公開鍵証明書の交換とは、メッセージの機密性のため、ＤＥＳセッション鍵の交換前または交換と同時に行われる。；

・セキュア・ローダは、データに対して完全性検査を行い、これが成功した場合にのみデータをインストールする。；

・このデータは、上述のプロトコルのうち１つを使用して実行される。；

・各開発者は、一般コンテンツ保護または特定コンテンツ保護のいずれも使用できる。

【００７６】

一形態としては：

・鍵Ｋを用いて暗号化されたデータは、Ｃの秘密コード署名鍵の下で署名され、Ｃによって、トラステッド・モジュールへ送信される。；

・Ｋに対応するロック解除鍵は、トラステッド・モジュールの公開鍵を用いてＣによって暗号化され、Ｃの秘密コード署名鍵を用いて署名され、コンピュータ・プラットフォームへ送信される。；

・鍵転送コードは、ロック解除鍵を復号化し、完全性と署名を検査し、その後、この鍵は、当該データと関連するトラステッド・モジュールに格納される。

【００７７】

別の形態としては：

・鍵Ｋを使用して暗号化されたデータは、Ｃの秘密コード署名鍵の下で署名され、Ｃによって、トラステッド・モジュールへ送信される。；

・ロック解除鍵は、Ｃからコンピュータ・プラットフォームのエンドユーザへ、またはコンピュータ・プラットフォームへ転送される。；

・鍵転送コードは、Ｋに対応する復号化鍵を、ロック解除鍵、トラステッド・モジュールまたはスマートカードのＩＤ、および事前に格納されているアルゴリズムから計算する。；

・任意で、前のステージは、そのデータと関連づけられたセキュア実行プログラムまたはソフトウェア実行プログラムにより実行される。；

・この復号化鍵は、その後、当該データと関連するトラステッド・モジュールまたはスマートカードに格納される。

【００７８】

さらなる形態としては：

・鍵Ｋと関連ソフトウェア実行プログラムを用いて暗号化されたデータは、Ｃの秘密コード署名鍵の下で署名され、Ｃによりトラステッド・モジュールへ送信される。；

・Ｋに対応するロック解除鍵は、トラステッド・モジュールＩＤまたはスマートカードＩＤに対応するデータベース・エントリに挿入される。

【００７９】

さらに別の形態としては：

・データおよび関連ソフトウェア実行プログラムは、Ｃの秘密コード署名鍵の下で署名され、Ｃによりトラステッド・モジュールへ送信される。；

・データを実行するための許可に対応するエントリが、トラステッド・モジュールＩＤまたはスマートカードＩＤに対応するデータベース・エントリ内へ挿入される。またはその逆が行われる。

【００８０】

本発明の特定の実施形態は、純粋に例として、添付の図面を参照することによ

って、ここで説明される。

【００８１】

本発明の実施形態を説明する前に、２０００年２月１５日付けの国際特許出願第ＰＣＴ／ＧＢ００／００５２８号の主題であるトラステッド（信頼）装置を取り入れたコンピューティング・プラットフォームが図１～図７を参照してまず説明される。この一般的な種類のコンピューティング・プラットフォームは、本発明の実施形態での使用に特に適している。また、コンピュータ・プラットフォームのユーザ専用のトラステッド・トークン装置の使用も説明される（後述の実施形態の一部に関連するため）－好適な実施例において、このトークン装置はスマートカードである。

【００８２】

この出願は、コンピューティングプラットフォームへの物理的なトラステッド装置またはモジュールの組み込みについて記載しており、これらの機能は、プラットフォームの識別を、プラットフォームの完全性メトリックを提供する信頼できる計測データに結び付け、それによって、「トラステッド・プラットフォーム」を形成している。識別および完全性メトリックは、プラットフォームの信頼性を保証することになっているトラステッド・パーティ（ＴＰ）により供給される期待値と比較される。適合が存在すれば、完全性メトリックの範囲に応じて、そのプラットフォームの少なくとも一部は正しく動作していることを意味している。

【００８３】

本明細書において、物理的または論理的コンポーネントに関連して使用された場合の「トラステッド」という用語は、その物理的または論理的コンポーネントが期待されたように常にふるまうことを意味するのに用いられる。そのコンポーネントのふるまいは予測可能であり、既知である。トラステッド・コンポーネントは、権限のない変更に対して高度の耐性を有する。

【００８４】

本明細書において、「コンピューティング・プラットフォーム」（またはコンピュータ・プラットフォーム」）とは、少なくとも１つのデータ・プロセッサと

少なくとも１つのデータ格納手段を指して用いており、これらは通常、複数のドライバなどの関連する通信ファシリティ、関連するアプリケーション及びデータファイルなどを有しているが、これらは本質的なものではなく、例えば、インターネットへの接続手段、外部ネットワークへの接続手段、または、ＣＤＲＯＭ、フロッピーディスク（Ｒ）、リボンテープなどのデータ記憶媒体に記憶されたデータの受信能力を有する入力ポートを持つことにより、ユーザまたは他のコンピュータプラットフォームなどの外部エンティティと対話することができる。

【００８５】

ユーザは、プラットフォームと他のデータを送受信する前にプラットフォームの正しい動作を検証する。ユーザは、トラステッド装置に、その識別と完全性メトリックを提供することを要求することによって、それを実行する。（任意で、トラステッド装置は、自身がプラットフォームの正しい動作を検証できない場合には、識別の証拠を提供することを拒否する）。ユーザは、識別と完全性メトリックの証明を受け取り、それらを正しいと信じる値に対して比較する。それらの適正値は、ユーザにより信用されるＴＰまたは他のエンティティにより供給される。トラステッド装置により報告されたデータがＴＰにより供給されたものと同じ場合、ユーザはそのプラットフォームを信用する。これは、ユーザがそのエンティティを信用するためである。このエンティティは、以前その識別を確認していて、そのプラットフォームの適正な完全性メトリックを判定しているため、プラットフォームを信用する。

【００８６】

例えば、コンピューティング・エンティティのユーザは、そのようなトラステッド・トークン装置の使用により、コンピュータ・エンティティに対して信用のレベルを設けることが可能である。トラステッド・トークン装置は、データ処理機能を有し、ユーザが高いレベルの信用を有する個人的および可搬式の装置である。それは、ユーザを識別するためにトラステッド・プラットフォームにより使用されることも可能である。このトラステッド・トークン装置は、以下の機能を実行することが可能である。：

・ユーザにとって直ぐに分かる方法で、コンピューティング・プラットフォーム

の正しい動作を検証する、例えば、音声または視覚表示による。；
・監視コンポーネントに、その監視コンポーネントが接続されるコンピュータ・プラットフォームの正しい動作の証明を提供するよう求める。；
・監視コンポーネントがコンピューティング・エンティティの正しい動作の満足な証拠を供給したか否かに応じて、また、正しい動作のそのような証拠がトークン装置により受信されない場合にコンピュータ・エンティティとの特定の対話を保留することにより、コンピューティング・プラットフォームとのトークン装置の対話のレベルを確立する。

【００８７】

ユーザがプラットフォームの信用できる動作を確立したら、ユーザはプラットフォームと他のデータを交換する。ローカル・ユーザの場合、この交換は、プラットフォーム上で動作するソフトウェア・アプリケーションとの対話により行うことができる。リモート・ユーザの場合、この交換は、セキュアなトランザクションを伴うものである場合がある。いずれの場合においても、交換されるデータはトラステッド装置によって「署名」される。それによって、ユーザは、データがそのふるまいが信用できるプラットフォームと交換されているという大きな自信を有する。

【００８８】

トラステッド装置は、暗号化処理を使用するが、そのような暗号化処理に対して外部インタフェースを提供する必要はない。また、最も望ましい実施は、トラステッド装置に不正防止機能を設け、これらを設けることにより、他のプラットフォーム機能から影響を受けないようにし、権限の無い変更に対して影響を受けない環境を提供することである。不正を防止することは不可能であるため、最良の近似は、不正耐性のあるまたは不正検出を行うトラステッド装置である。したがって、トラステッド装置は、不正耐性のある１つの物理的コンポーネントから成ることが好ましい。

【００８９】

不正耐性に関する技術は、セキュリティの当業者には周知である。これらの技術には、不正に抵抗する方法（たとえばトラステッド装置の適切なカプセル化な

ど）、改ざんを検出する方法（たとえば所定の電圧からの外れの検出、Ｘ線、またはトラステッド装置ケーシング内の物理的な完全性の欠落の検出など）、及び、不正が検出されたときにデータを削除する方法が含まれる。適切な技術のさらなる説明は、ｈｔｔｐ：／／ｗｗｗ．ｃｌ．ｃａｍ．ａｃ．ｕｋ／～ｍｇｋ２５／ｔａｍｐｅｒ．ｈｔｍｌで見られる。不正防止は説明されるシステムの最も望ましい機能であるが、本発明の通常の動作には入らない、したがって、それは本発明の範囲外であり、ここではこれ以上詳細に説明されない。

【００９０】

トラステッド装置は、偽造が困難でなければならないため、物理的な装置であることが好ましい。偽造が困難でなければならないため、不正耐性があることが最も好ましい。トラステッド装置は、一般に、ローカルと遠隔の両方において、素性を証明する必要があるため、暗号化処理を利用できるエンジンを有し、また、トラステッド装置が関連づけられるプラットフォームの完全性メトリックを計測するための少なくとも１つの手段を有する。

【００９１】

トラステッド・プラットフォーム１０は、図１の略図に示される。プラットフォーム１０は、キーボード１４（ユーザの確認キーを提供する）、マウス１６、および、モニター１８からなる標準的な機能を備え、これらはプラットフォームの物理的な「ユーザインタフェース」を提供する。また、トラステッド・プラットフォームのこの実施形態は、スマートカードリーダ１２も含む。スマートカードリーダ１２の横に、信用できるユーザが、後述のトラステッド・プラットフォームと対話できるようにスマートカード１９が図示されいる。プラットフォーム１０において、複数のモジュール１５がある。それらは、そのプラットフォームに適した実質的に任意の種類のトラステッド・プラットフォームの他の機能的要素である。そのような要素の機能的な意味は、本発明では問題としていないため、ここでこれ以上説明は行わない。トラステッド・コンピュータ・エンティティの付加的なコンポーネントには、一般に、１つ以上のローカル・エリア・ネットワーク（ＬＡＮ）ポート、１つ以上のモデム・ポート、及び、１つ以上の電源、冷却ファン等が含まれる。

【００９２】

図２に示されるように、トラステッド・コンピューティング・プラットフォーム１０のマザーボード２０には、（標準コンポーネントの内でも）メインプロセッサ２１、メインメモリ２２、トラステッド装置２４、データ・バス２６、およびそれぞれの制御線２７および回線２８、プラットフォーム１０のＢＩＯＳプログラムを含むＢＩＯＳメモリ２９、および、マザーボードのコンポーネントと、スマートカードリーダ１２、キーボード１４、マウス１６及びモニター１８（さらにモデム、プリンター、スキャナーなどの他の周辺装置）との間で対話を制御する入出力（ＩＯ）装置２３、が含まれる。メインメモリ２２は、一般に、ランダム・アクセス・メモリ（ＲＡＭ）である。動作時には、プラットフォーム１０は、たとえばウィンドウズＮＴ（商標）などのオペレーティング・システムを、ハードディスク（図示せず）からＲＡＭ内へロードする。さらに、動作時において、プラットフォーム１０は、プラットフォーム１０により実行される処理またはアプリケーションを、ハードディスク（図示せず）からＲＡＭ内へロードする。

【００９３】

コンピュータ・エンティティは、物理的アーキテクチャだけでなく論理的アーキテクチャをも有すると考えられる。この論理的アーキテクチャは、コンピュータ・プラットフォームとトラステッド・コンポーネントとの間において、図１～図４に記載の物理的アーキテクチャと同じ基本的な区分を有している。すなわち、トラステッド・コンポーネントは物理的に関連するコンピュータ・プラットフォームから論理的に分離している。コンピュータ・エンティティは、コンピュータ・プラットフォーム（第１位置のプロセッサと第１のデータ格納手段）上に物理的に常駐する論理空間であるユーザ空間と、トラステッド・コンポーネント上に物理的に常駐する論理空間であるトラステッド・コンポーネント空間とから成る。ユーザ空間には、１つまたは複数のドライバ、１つまたは複数のアプリケーション・プログラム、ファイル格納領域、スマートカード読取プログラム、スマートカード・インタフェース、及び、ユーザ空間において動作を実行しトランステッド・コンポーネントへ報告を返すソフトウェア・エージェントが存在する。

トラステッド・コンポーネント空間は、トラステッド・コンポーネントの第２のデータ・プロセッサ及び第２のメモリ領域によりサポートされるトラステッド・コンポーネントに基づく論理領域であり、トラステッド・コンポーネント内に物理的に常駐する論理領域である。モニター１８は、トラステッド・コンポーネント空間から直接画像を受信する。コンピュータ・エンティティの外部には、インターネット、さまざまなローカル・エリア・ネットワーク、ドライバ（１つ以上のモデム・ポートを備えることが可能）を介してユーザ空間に接続されるワイドエリア（広域）ネットワークなどの外部通信ネットワークがある。外部のユーザ・スマートカードは、ユーザ空間のスマートカード読取プログラムへ入力する。

【００９４】

一般に、パーソナル・コンピュータにおいて、ＢＩＯＳプログラムは、特別に予約されたメモリ領域に配置され、最初の１メガバイトうち上位６４ＫはシステムＲ・メモリ（アドレスＦＯＯＯｈ～ＦＦＦＦｈ）の役割を果たし、メインプロセッサは、業界全体標準にしたがって、このメモリ位置を最初に参照するように構成される。

【００９５】

本プラットフォームと従来のプラットフォームとの大きな違いは、リセット後に、メインプロセッサはトラステッド装置により最初に制御され、その後、トラステッド装置がそのプラットフォーム専用のＢＩＯＳプログラムへ制御を渡し、次に、このＢＩＯＳプログラムが通常通りすべての入出力装置を初期化する。ＢＩＯＳプログラムが実行された後、制御は、ＢＩＯＳプログラムにより、通常通りウィンドウズＮＴ（商標）など、ハードディスク・ドライブ（図示せず）からメインメモリ２２に典型的にロードされるオペレーティング・システムプログラムへ渡される。

【００９６】

当然ながら、この通常の処理からの変更は、業界標準の実施への修正が必要であり、それによって、メインプロセッサ２１は、トラステッド装置２４がメインプロセッサの最初の命令を受信するアドレスへ向けられる。この変更は、単に、様々なアドレスをメインプロセッサ２１内にハード・コーディングすることによ

り成される。または、トラステッド装置２４に、標準ＢＩＯＳプログラム・アドレスを割り当てることにより成される。この場合、メインプロセッサの構成を修正する必要はない。

【００９７】

ＢＩＯＳブート・ブロックが、トラステッド装置２４内に含まれることは非常に望ましい。これは、完全性メトリックの取得の破壊（もしくは、悪質なソフトウェア処理が存在する場合に起こり得る）を防止し、ＢＩＯＳ（正しい場合でも）がオペレーティング・システムに対して適性な環境を構築できない状況を創り出す悪質なソフトウェア処理を防止する。

【００９８】

本明細書で説明されるシステムにおいて、トラステッド装置２４は単一の分離されたコンポーネントであるが、トラステッド装置２４の機能は、マザーボード上の複数の装置上へ分割することも可能であり、あるいは、プラットフォーム上の１つ以上の既存の標準装置内へ集積することさえ可能であると想像される。例えば、トラステッド装置の１つ以上の機能を、それらの機能及び通信が破壊されることの無いメインプロセッサ自身内へ集積させることも可能である。しかしながら、この場合、プロセッサがトラステッド機能により単独使用されるためには、おそらくプロセッサ上に個別のリードが必要とされる。さらに又はあるいは、本発明においてトラステッド装置はマザーボード２０への統合のために改造されたハードウェア装置であるが、トラステッド装置は、ドングルなどのように、必要な時にプラットフォームへ装着される「着脱可能な」装置として実施することも可能である。トラステッド装置が統合されるか着脱可能であるかは、設計上の選択の問題である。しかし、トラステッド装置が分離されている場合は、トラステッド装置とプラットフォームとの間に論理的結合を提供する機構が存在しなければならない。

【００９９】

図３に示すように、トラステッド装置２４は、数多くのブロックを含む。システムがリセットされた後、トラステッド装置２４は、プラットフォーム１０のオペレーティング・システム（システム・クロックとモニター上の表示を含む）が

適切かつセキュアな方法で動作していることを保証にするため、セキュア・ブート処理を実行する。セキュア・ブート処理の間、トラステッド装置２４は、コンピューティング・プラットフォーム１０の完全性メトリックを取得する。また、トラステッド装置２４は、セキュア・データ転送を実施することも可能であり、例えば、暗号化／復号化及び署名／検証によって、トラステッド装置とスマートカード間の認証を実施することができる。また、トラステッド装置２４は、ユーザ・インタフェースのロックなど、さまざまなセキュリティ制御ポリシーをセキュアに実施することもできる。

【０１００】

具体的には、トラステッド装置は、トラステッド装置２４の全体的な動作を制御し、トラステッド装置２４上の他の機能及びマザーボード２０上の他の装置と対話するようにプログラムされた制御装置３０と、プラットフォーム１０から完全性メトリックを取得するための計測機能３１と、指定されたデータを署名、暗号化または復号化するための暗号化機能３２と、スマートカードを認証するための認証機能３３と、マザーボード２０のデータ・バス２６、制御線２７及びアドレス線２８のそれぞれにトラステッド装置２４を接続するための適切なポート（３６、３７、３８）を有するインタフェース回路（サーキットリー）３４とを含む。トラステッド装置２４のそれぞれのブロックは、トラステッド装置２４の適切な揮発性メモリ領域４、及び／又は、不揮発性メモリ領域３への（典型的に制御装置３０を介して）アクセスを有する。さらに、トラステッド装置２４は、不正耐性を得るように、既知の方法で設計されている。

【０１０１】

性能上の理由により、トラステッド装置２４は、アプリケーションに特化した集積回路（ＡＳＩＣ）として実施される。しかしながら、柔軟性を持たせるためには、トラステッド装置２４は、適切にプログラムされたマイクロ・コントローラであることが好ましい。ＡＳＩＣ及びマイクロ・コントローラは、マイクロエレクトロニクスの業界では周知のものであるため、本明細書においてこれ以上詳細には説明されない。

【０１０２】

トラステッド装置２４の不揮発性メモリ３内に格納されたデータの一項目は、証明書３５０である。証明書３５０は、少なくとも、トラステッド装置２４の公開鍵３５１、及び、トラステッド・パーティ（ＴＰ）により計測されるプラットフォーム完全性の認証値３５２を含んでいる。証明書３５０は、トラステッド装置２４に格納される前に、ＴＰの秘密鍵を使用してＴＰにより署名される。後の通信セッションにおいて、プラットフォーム１０のユーザは、取得した完全性メトリックを正規の完全性メトリック３５２と比較することによって、プラットフォーム１０の完全性を検証することができる。合致があれば、ユーザはプラットフォーム１０が破壊されなかったことを確信できる。ＴＰの一般利用可能な公開鍵の情報によって、証明書３５０の簡単な検証が可能となる。不揮発性メモリ３５は、識別（ＩＤ）ラベル３５３も含む。ＩＤラベル３５３は、例えばシリアル番号などの従来のＩＤラベルであり、何らかのコンテクスト内で一意なものである。ＩＤラベル３５３は、一般に、トラステッド装置２４に関連するデータの索引およびラベル付けのために使用されるが、信用できる条件下においてプラットフォーム１０の識別を証明するには、それ自身だけでは不十分である。

【０１０３】

トラステッド装置２４には、関連するコンピューティング・プラットフォーム１０の完全性メトリックを確実に計測または取得する少なくとも１つの方法が設けられる。本実施形態において、完全性メトリックは、計測機能３１がＢＩＯＳメモリ内のＢＩＯＳ命令のダイジェストを生成することにより取得される。このようにして取得された完全性メトリックは、上述のように検証されると、このプラットフォーム１０がハードウェアまたはＢＩＯＳプログラムのレベルで破壊されていないことを示す高レベルの信用を、プラットフォーム１０を利用する可能性のあるユーザに提供する。オペレーティングシステム及びアプリケーション・プログラム・コードが破壊されていないことを検査するため、一般に、他の周知のプロセス、例えばウィルスチェッカーなども設けられるであろう。

【０１０４】

計測機能３１は、トラステッド装置２４のハッシュ・プログラム３５４及び秘密鍵３５５を格納する不揮発性メモリ３と、取得された完全性メトリックをダイ

ジェスト３６１の形で格納する揮発性メモリ４に対してアクセスを有する。また、適切な実施形態において、揮発性メモリ４は、プラットフォーム１０へのアクセス権を獲得するために用いられる１つ以上の認証スマートカード１９の公開鍵及び関連するＩＤラベル３６０ａ～３６０ｎを格納するのにも用いられる。

【０１０５】

一好適実施例においては、ダイジェストだけでなく完全性メトリックもブール値を含んでおり、これらは、後に明らかにされる理由により、計測機能３１によって揮発性メモリ４に格納される。

【０１０６】

図４を参照し、完全性メトリックを取得するための好適な処理を説明する。

【０１０７】

ステップ4.００において、スイッチがオンされると、計測機能３１は、トラステッド装置２４がアクセスされる第１のメモリであるか否かを判定するため、データ線、制御線及びアドレス線（２６，２７及び２８）上において、メインプロセッサ２１のアクティビティを監視する。従来の動作では、メインプロセッサは、ＢＩＯＳプログラムを実行するため、最初にＢＩＯＳメモリへ向けられていた。しかしながら、本実施形態によれば、メインプロセッサ２１は、メモリとして機能するトラステッド装置２４へ向けられる。ステップ４０５において、トラステッド装置２４がアクセスされた最初のメモリである場合は、ステップ４１０において、計測機能３１が、トラステッド装置２４がアクセスされた最初のメモリであったことを示すブール値を不揮発性メモリ３へ書き込む。そうでない場合は、ステップ４１５において、計測機能が、トラステッド装置２４がアクセスされる最初のメモリではなかったことを示すブール値を書き込む。

【０１０８】

トラステッド装置２４が最初にアクセスされたものでないというイベントには、もちろん、トラステッド装置２４がまったくアクセスされない可能性も存在する。これは、たとえば、メインプロセッサ２１がＢＩＯＳプログラムを最初に実行するように操作された場合などである。このような状況下では、プラットフォームは動作するが、その完全性を要求に応じて検証することはできないであろう

。なぜなら、完全性メトリックが利用できないからである。さらに、ＢＩＯＳプログラムがアクセスされた後にトラステッド装置２４がアクセスされると、このブール値は、プラットフォームの完全性の欠如を明確に示すであろう。

【０１０９】

ステップ４２０において、メインプロセッサ２１によってメモリとしてアクセスされると、ステップ４２５において、メインプロセッサ２１は、計測機能３１から格納されたネイティブのハッシュ命令３５４を読み込む。このハッシュ命令３５４は、メインプロセッサ２１による処理のため、データバス２６を介して受け渡される。ステップ４３０において、メインプロセッサ２１はこのハッシュ命令３５４を実行し、ステップ４３５においてこれらを用い、ＢＩＯＳメモリ２９のコンテンツを読み込み、ハッシュ・プログラムによりこれらのコンテンツを処理することによって、ＢＩＯＳメモリ２９のダイジェストを計算する。ステップ４４０において、メインプロセッサ２１は、トラステッド装置２４内の適切な不揮発性メモリ位置４へ計算されたダイジェスト３６１を書き込む。次に、ステップ４４５において、計測機能３１は、ＢＩＯＳメモリ２９内のＢＩＯＳプログラムを呼び出し、従来の方法で実行が継続される。

【０１１０】

明らかに、完全性メトリックを計算する方法は、必要とされる信用の範囲に応じて多数の異なる方法が存在する。ＢＩＯＳプログラムの完全性の計測は、プラットフォームの基礎を成す処理環境の完全性について、基本的な検査を提供する。完全性メトリックは、ブート処理の正当性について推測を可能とする形態でなければならない。すなわち、完全性メトリックの値は、正しいＢＩＯＳを使用してブートられたプラットフォームであるか否かを検証できるものでなければならない。任意であるが、ＢＩＯＳ内の個々の機能ブロックは、自体のダイジェスト値を、これら個々のダイジェストのダイジェストである総体的ＢＩＯＳダイジェストと共に有することもできる。これによって、ＢＩＯＳ動作のいずれの部分が意図する目的について重要であるか、及び、いずれの部分が問題とされないかということを、ポリシーに明示することが可能になる（この場合、個々のダイジェストは、そのポリシーの下で、動作の正当性を立証できるような方法で格納され

なければならない）。

【０１１１】

他の完全性検査には、プラットフォームに取り付けられた様々な他の装置、コンポーネントまたは機器が存在し、正しい動作順位であることの検証が含まれる。一例としては、ＳＣＳＩコントローラに関連するＢＩＯＳプログラムは、周辺機器との通信が信用できることを保証するため、検証される場合がある。別の例としては、他の装置、例えばプラットフォーム上のメモリ装置またはコプロセッサは、安定した結果を保証するために、固定のチャレンジ／レスポンスの対話を規定することにより、検証される場合がある。ここで、トラステッド装置２４が個別のコンポーネントである場合、何らかのそのような会話の形態は、トラステッド装置２４とプラットフォームの間に適切な論理的結合を提供することが望ましい。また、本実施の形態において、トラステッド装置２４は、プラットフォームの他の部分との通信の主要な手段としてデータ・バスを利用しているが、ハードワイヤード・パスまたは光学式パスなど、代替の通信路を設けることも便利ではないが実現可能である。さらに、本発明の実施形態において、トラステッド装置２４は、完全性メトリックをメインプロセッサ２１に計算させているが、別の実施形態として、トラステッド装置自体が、１つ以上の完全性メトリックを計測するように構成されることも可能である。

【０１１２】

ＢＩＯＳブート処理は、そのブート処理自体の完全性を検証するための機構を有することが好ましい。そのような機構は、インテルのドラフト「Ｗｉｒｅｄ ｆｏｒ Ｍａｎａｇｅｍｅｎｔ ｂａｓｅｌｉｎｅ ｓｐｅｃｉｆｉｃａｔｉｏｎ ｖ ２．０－ＢＯＯＴ Ｉｎｔｅｇｒｉｔｙ Ｓｅｒｖｉｃｅ」などからすでに周知であり、ソフトウェアまたはファームウェアをロードする前に、そのソフトウェアまたはファームウェアのダイジェストを計算することが含まれている。そのような計算されたダイジェストは、公開鍵がＢＩＯＳに知らされているトラステッド・エンティティにより供給される証明書に格納された値と比較される。その後、計算された値が証明書から得られた期待値と適合した場合にのみ、ソフトウェア／ファームウェアがロードされ、また、証明書は、トラステッド・エ

ンティティの公開鍵を用いて正当性が証明されている。そうでない場合は、適切な例外処理ルーチンが呼び出される。

【０１１３】

任意であるが、計算されたＢＩＯＳダイジェストを受信した後、トラステッド装置２４は証明書内のＢＩＯＳダイジェストの適切な値を検査し、計算されたダイジェストが適切な値と適合しない場合、ＢＩＯＳに制御を渡さないことも可能である。さらに又はもしくは、トラステッド装置２４は、ブール値を検査し、トラステッド装置２４がアクセスされる最初のメモリでなかった場合には、制御をＢＩＯＳへ返さないことが可能である。これらの場合のいずれにおいても、適切な例外処理ルーチンが実行されることが可能である。

【０１１４】

図５には、ＴＰによるアクションのフローと、プラットフォームに組み込まれたトラステッド装置２４と、トラステッド・プラットフォームの完全性を検証したい（リモート・プラットフォームの）ユーザとが示されている。ユーザがローカル・ユーザの場合には、図５に示されるものとほぼ同じステップが実行されることは明らかであろう。いずれの場合においても、ユーザは、一般に、検証を規定するための何らかのソフトウェア・アプリケーションに依存するであろう。リモート・プラットフォームまたはトラステッド・プラットフォーム上でこのソフトウェア・アプリケーションを動作させることは可能であろう。しかしながら、リモート・プラットフォーム上でさえ、ソフトウェア・アプリケーションは何らかの方法で破壊される可能性がある。したがって、高レベルの完全性を求めるには、このソフトウェア・アプリケーションがユーザのスマートカード上に存在し、ユーザが検証の目的で適切なリーダへこのスマートカードを挿入することが好ましい。特定の実施形態はこのような構成に関連している。

【０１１５】

最初のステップとして、トラステッド・プラットフォームを保証するＴＰは、それを保証するか否かを判定するためにプラットフォームの種類を調べる。これは、ポリシーの問題であろう。すべてが良好であれば、ステップ５００において、ＴＰは、プラットフォームの完全性メトリックの値を計測する。次に、ステッ

プ５０５において、ＴＰは、そのプラットフォームに対して証明書を生成する。この証明書は、トラステッド装置の公開鍵及び任意でそのＩＤラベルを計測された完全性メトリックに添付し、この文字列をＴＰの秘密鍵を用いて署名することによって、ＴＰにより作成される。

【０１１６】

続いて、トラステッド装置２４は、その秘密鍵を用いてユーザから受信した入力データを処理し、その秘密鍵の知識なく入力／出力の組を生成することが統計的に不可能であるように出力データを生成することによって、その識別を証明することができる。したがって、この場合、この秘密鍵が識別の基礎を形成する。当然ながら、対称鍵暗号を用いて識別の基礎を形成することも可能ではある。しかしながら、対称鍵暗号を用いることの欠点は、ユーザが、自分の秘密をトラステッド装置と共有する必要があるということである。さらに、ユーザと秘密を共有する必要が生じる結果として、対称鍵暗号は、ユーザへ識別の証明を行うには十分であるが、トラステッド装置またはユーザからの検証を完全に信用できない第三者へ識別の証明を行うには不充分である。

【０１１７】

ステップ５１０において、トラステッド装置２４は、トラステッド装置２４の適切な不揮発性メモリ位置３に証明書３５０を書き込むことによって初期化される。これは、トラステッド装置がマザーボード２０にインストールされた後、トラステッド装置２４とのセキュアな通信により成されることが好ましい。トラステッド装置２４へ証明書を書き込む方法は、秘密鍵をスマートカードに書き込むことによりスマートカードを初期化する際に用いられる方法に類似している。セキュアな通信は、ＴＰにのみ知られる「マスターキー」によってサポートされる。このマスターキーは、製造時にトラステッド装置（またはスマートカード）へ書き込まれており、トラステッド装置２４へデータの書き込みを可能にするのに用いられる。したがって、このマスターキーの知識無くしてトラステッド装置へのデータの書き込みは不可能である。

【０１１８】

プラットフォーム動作時のいくらか後の時点において、例えばステップ５１５

においてスイッチがオンされた、またはリセットされたとき、トラステッド装置２４は、そのプラットフォームの完全性メトリック３６１を取得して格納する。

【０１１９】

ユーザがプラットフォームと通信したい場合、ステップ５２０において、ユーザは、乱数などのノンスを作成し、ステップ５２５において、トラステッド装置２４にチャレンジを送る（プラットフォームのオペレーティング・システム、または適切なソフトウェア・アプリケーションは、このチャレンジを認識し、典型的にはＢＩＯＳタイプの呼び出しを介して、これを適切な方法でトラステッド装置２４へ渡すように構成される）。このノンスは、信用できないプラットフォームによる古いが真性の署名の繰り返し（「再生攻撃」と呼ばれる）によって、発生する詐欺からユーザを保護するために使用される。ノンスを供給しその応答を検証する処理は、周知の「チャレンジ／レスポンス」処理の例である。

【０１２０】

ステップ５３０において、トラステッド装置２４はチャレンジを受信し、適切な応答を生成する。応答は、計測された完全性メトリックおよびノンスのダイジェストと、任意でそのＩＤラベルを含む。次に、ステップ５３５において、トラステッド装置２４は、自身の秘密鍵を使用してダイジェストに署名し、この署名されたダイジェストを証明書３５０と共にユーザへ返す。

【０１２１】

ステップ５４０において、ユーザはチャレンジのレスポンスを受信し、既知であるＴＰの公開鍵を用いて証明書を検証する。次に、ステップ５５０において、ユーザは証明書からトラステッド装置２４の公開鍵を取り出し、これを用いてチャレンジのレスポンスから得られた署名されたダイジェストを復号する。次に、ステップ５６０において、ユーザはチャレンジのレスポンス内のノンスを検証する。次に、ステップ５７０において、ユーザは、計算された完全性メトリック即ちチャレンジのレスポンスから取り出した完全性メトリックを、適切なプラットフォームの完全性メトリック即ち証明書から取り出された完全性メトリックと比較する。ステップ５４５，５５５，５６５，または５７５において前述の検証ステップのいずれかが失敗した場合、全処理はステップ５８０で終了し、それ以上

通信は行われない。

【０１２２】

ステップ５８５及び５９０において、すべてが良好であると仮定すると、ユーザ及びトラステッド・プラットフォームは、他のプロトコルを用いて他のデータのためのセキュアな通信を準備する。この場合、プラットフォームからのデータはトラステッド装置２４によって署名されること好ましい。

【０１２３】

この検証処理のさらなる改良も可能である。チャレンジの送信者は、そのチャレンジにより、プラットフォームの完全性メトリックの値とそれが得られた方法との両方を知ることが好ましい。これら両方の情報により、チャレンジの送信者がそのプラットフォームの完全性について適切な判定を行えるようにすることが望ましい。チャレンジの送信者は多数の様々なオプションを利用することも可能であり、例えば、トラステッド装置において完全性メトリックが適正であると認識されたことを受け入れるオプション、または、完全性メトリックの値がチャレンジを送るユーザにより保持される値と等しい場合にのみ、そのプラットフォームが当該レベルを有することを受け入れるするオプション（または、これら２つの場合について異なる信用のレベルを保持することができるオプション）などが利用できる。

【０１２４】

署名、証明書の使用、及び、チャレンジ／レスポンスの技術、並びに、これらを利用して識別を証明する技術は、セキュリティの当業者によって周知であるため、ここではこれ以上詳しく説明する必要はないであろう。

【０１２５】

ユーザのスマートカード１９は、コンピューティング・エンティティとは分離されたトークン装置であり、スマートカードリーダポート１９を介してコンピューティング・エンティティと対話する。ユーザは、いくつかの異なるベンダまたはサービスプロバイダにより発行されるいくつかの異なるスマートカードを有することが可能であり、本明細書に記載されるようなトラステッド・コンポーネント及びスマートカードリーダが設けられたコンピューティング・エンティティの

うちの任意の１つから、インターネットまたは複数のネットワークコンピュータへのアクセスを得ることが可能である。ユーザが使用している特定のコンピューティング・エンティティにおけるユーザの信用は、ユーザのトラステッド・スマート・カード・トークンとそのコンピューティング・エンティティのトラステッド・コンポーネントとの間に対話から導出される。ユーザは、ユーザのトラステッド・スマート・カード・トークンに基づいて、トラステッド・コンポーネントの信用度を検証
する。

【０１２６】

ユーザ・スマートカード１９の処理部６０が、図６に示されている。図示のように、ユーザ・スマートカード１９の処理部６０は、プロセッサ６１、メモリ６２、及び、インタフェース・コンタクト６３の標準的な機能を有する。後述のように、プロセッサ６１は、ユーザ・スマートカード１９の認証、及び、プラットフォーム１０の検証を含む簡単なチャレンジ／レスポンス動作のためにプログラムされている。メモリ６２は、その秘密鍵６２０、その公開鍵６２８、（任意で）ユーザ・プロファイル６２１、ＴＰの公開鍵６２２、及び、識別６２７を含んでいる。ユーザ・プロファイル６２１は、ユーザにより利用可能な予備のスマートカード２０　ＡＣ１～ＡＣｎ、及び、そのユーザについてのセキュリティ・ポリシー６２４を一覧にしている。ユーザ・プロファイルは、予備のスマートカード２０のそれぞれに対して、識別情報６２３、スマートカード間のトラステッド構造６２５（もし存在するなら）、及び、（任意で）スマートカードの種類または形式６２６を含んでいる。

【０１２７】

ユーザ・プロファイル６２１において、補助スマートカード２０エントリＡＣ１～ＡＣｎのそれぞれには、カードの種類により異なる関連識別情報６２３が含まれる。たとえば、キャッシュカードの識別情報には、一般に、単純なシリアル番号が含まれており、暗号カードの識別情報には、一般に、暗号カードの公開鍵（または証明書）が含まれている（秘密鍵は、暗号カード自体に秘密として保存される）。

【０１２８】

「セキュリティ・ポリシー」６２４は、補助スマートカード２０を使用している間に、ユーザがプラットフォーム１０上に有する許可を示している。たとえば、補助スマートカード２０が使用されている間、ユーザ・インタフェースは、補助スマートカード２０の機能に応じてロックまたはロック解除される。

【０１２９】

さらに又はもしくは、プラットフォーム１０上の特定のファイルまたは実行プログラムは、特定の補助スマートカード２０がどれくらい信用できるものかに応じて、アクセスの可能または不可が決定される。さらに、セキュリティ・ポリシー６２４は、後述されるように、「クレジット・レシート」または「一時的委任」など、補助スマートカード２０に対して特定の動作モードを指定できる。

【０１３０】

「トラスト構造」６２５は、ユーザ・スマートカード１９を最初に再使用することなく、補助スマートカード２０がそれ自体でさらなる補助スマートカード２０をシステムに「導入する」ことができるか否かを定義する。本明細書で詳述される実施例において、ユーザ・スマートカード１９とユーザ・スマートカード１９によりプラットフォームへ導入することの可能な補助スマートカード２０との間には、定義されたトラスト構造のみが存在する。導入には、後述のように「単一セッション」または「複数セッション」が在り得る。しかしながら、特定の補助スマートカード２０は、実際には、さらなる補助スマートカード２０を導入することもできる。この場合、補助スマートカード２０は、導入することが可能な各補助スマートカードの一覧が記載されたユーザ・プロファイルに相当する物を有する必要がある。補助スマートカード２０の使用は、本発明の必須な特徴ではないため、本願ではこれ以上説明されない。補助スマートカードの使用は、本出願人の同時係属出願である「Ｃｏｍｐｕｔｉｎｇ　Ａｐｐａｒａｔｕｓ　ａｎｄ　Ｍｅｔｈｏｄｓ　ｏｆ　Ｏｐｅｒａｔｉｎｇ　Ｃｏｍｐｕｔｉｎｇ　Ａｐｐａｒａｔｕｓ」と題する２０００年３月５日付の国際特許出願第ＰＣＴ／ＧＢ００／００７５１号（本明細書において、引用されている）の主題である。

【０１３１】

ここで、ユーザ・スマートカード１９とプラットフォーム１０との間の認証のための好適な処理について、図７の系統線図を参照して説明する。これから説明するが、この処理は、チャレンジ／レスポンスルーチンを不都合なく実施するものである。利用可能なチャレンジ／レスポンス機構は多数存在する。本実施例で使用される認証プロトコルの実施形態は、ＩＳＯ／ＩＥＣ９７９８－３に記載されているような、相互（または３ステップの）認証である。もちろん、他の認証手順、例えば同様にＩＳＯ／ＩＥＣ９７９８－３に記載される２ステップまたは４ステップなどを用いることも可能である。

【０１３２】

初めに、ステップ７００において、ユーザは、ユーザ・スマートカード１９をプラットフォーム１０のスマートカードリーダ１２へ挿入する。前もって、プラットフォーム１０は、一般に、標準オペレーティング・システムの制御下で動作させられ、認証プロセスを実行しているため、ユーザがユーザ・スマートカード１９を挿入するのを待機している。このようにしてスマートカードリーダ１２がアクティブにされる一方、プラットフォーム１０は、一般に、ユーザ・インタフェース（即ち、画面、キーボード及びマウス）を「ロック」することによりユーザに対してアクセス不能にされる。

【０１３３】

ユーザ・スマートカード１９が、スマートカードリーダ１２に挿入されると、ステップ７０５において、トラステッド装置２４は、これをトリガにしてノンスのＡを生成し、ユーザ・スマートカード１９へ送信することにより相互認証を試みる。信用の無い第三者による古いが真性の応答の反復により行われる不正（「再生攻撃」と呼ばれる）から発信者を保護するため、乱数などのノンスが使用される。

【０１３４】

それに応答して、ステップ７１０において、ユーザ・スマートカード１９は、ノンスＡのプレーンテキスト、ユーザ・スマートカード１９により生成されたノンスＢ、及び、トラステッド装置２４のＩＤ３５３からなる結合と、プレーンテキストをユーザ・スマートカード１９の秘密鍵で署名して生成したプレーンテキ

ストの署名と、ＩＤを含む証明書と、スマートカード１９の公開鍵とを含む応答を生成して返す。

【０１３５】

ステップ７１５において、トラステッド装置２４は、証明書に含まれる公開鍵を用いて前記プレーンテキストの署名を検証することにより、この応答を認証する。応答が認証されなかった場合、処理はステップ７２０において終了する。応答が認証された場合、ステップ７２５において、トラステッド装置２４は、ノンスＡのプレーンテキスト、ノンスＢ、ユーザ・スマートカード１９のＩＤ６２７、及び、取得された完全性メトリックからなる結合と、トラステッド装置２４の秘密鍵を用いてプレーンテキストを署名することにより生成されたプレーンテキストの署名と、ＴＰの秘密鍵を用いて署名されたトラステッド装置２４の公開鍵及び正規の完全性メトリックを含む証明書とを含むさらなる応答を生成して送信する。

【０１３６】

ユーザ・スマートカード１９は、この応答にＴＰの公開鍵を用い、取得された完全性メトリックを正規の完全性メトリックと比較することによりこの応答を認証する。ステップ７３０において、比較の「適合」は認証の成功を意味しており、もしこのさらなる応答が認証されなかった場合には、処理はステップ７３５で終了する。

【０１３７】

認証が成功した場合、トラステッド装置２４はユーザ・スマートカード１９の認証を完了し、かつ、ユーザ・スマートカード１９はトラステッド・プラットフォーム１０の完全性の検証を完了し、ステップ７４０において、認証処理は、ユーザに対しセキュア処理を実行する。そして、ステップ７４５において、認証処理は、インターバル・タイマーを設定する。その後、ステップ７５０において、認証処理は、所定のタイムアウト時間に一致または超過したか否かを周期的に検出するため、適切なオペレーティングシステム割り込みルーチンを用いて、インターバル・タイマーをサービスする。

【０１３８】

当然ながら、認証処理およびインターバル・タイマーは、セキュア処理と平行して動作する。タイムアウト時間になると、またはそれを過ぎると、ステップ７６０において、認証処理は、これをトリガにして、トラステッド装置２４にユーザ・スマートカード１９へ自身を識別させるためのチャレンジを送信させることにより、ユーザ・スマートカード１９を再認証させる。ステップ７６５において、ユーザ・スマートカード１９は、自身のＩＤ６２７と自身の公開鍵６２８を含む証明書を返す。ステップ７７０において、応答がまったくない場合（例えば、ユーザ・スカートカード１９が取り除かれてしまった場合など）、または証明書が何らかの理由で有効でなくなった場合（例えば、ユーザ・スマートカードが別のスマートカードに置換されてしまった場合など）、そのセッションは、ステップ７７５においてトラステッド装置２４により終了される。そうでない場合、ステップ７７０において、ステップ７４５からの処理は、インターバル・タイマーをリセットすることによって繰り返される。

【０１３９】

署名、証明書の使用、及び、チャレンジ／レスポンスの技術、並びに、これらを利用して識別を証明する技術は、セキュリティ当業者にとって周知であるため、本明細書においてこれ以上詳述されない。

【０１４０】

次に、図１～図７を参照して上述されたシステムの修正形態を、図２１および図８～図１３を参照して説明する。この修正形態は、２０００年２月１５日付の国際特許出願第ＰＣＴ／ＧＢ００／００５０４号の主題である。図２１において、ホスト・コンピュータ１００は、主ＣＰＵ１０２、ハードディスク・ドライブ１０４、ＰＣＩネットワーク・インタフェース・カード１０６、及び、ＤＲＡＭメモリ１０８を有し、これらの間に従来の（「通常の」）通信路１１０（ＩＳＡ，ＥＩＳＡ，ＰＣＩ，ＵＳＢなど）を備える。また、ネットワーク・インタフェース・カード１０６は、ホスト・コンピュータ１００の外界との外部通信路１１２も有する。

【０１４１】

ネットワーク・インタフェース・カード１０６は、「赤」および「黒」のデー

タ領域１１４，１１６へ論理的に分割され、これらの間にインタフェース１１８を有する。赤領域１１４では、データは、通常プレーン・テキストであり、検出不可能な変更と望ましくない盗聴に対して敏感で攻撃を受けやすい。黒データ領域１１６では、データは、検出されない変更と望ましくない盗聴から保護されている（標準的な暗号機構によって、暗号化されることが好ましい）。インタフェース１１８は、赤の情報が黒領域１１６へリークしないことを保証している。赤領域１１４と黒領域１１６を分離するため、インタフェース１１８は、標準的な暗号化方法及び電子分離技術を用いることが好ましい。このような赤領域１１４、黒領域１１６及びインタフェース１１８の設計と構築は、特に軍事分野などのセキュリティおよび電子工学業界の当事者にとって周知である。通常の通信路１１０及び外部通信路１１２は、ネットワーク・インタフェース・カード１０６の黒領域１１６と接続している。

【０１４２】

また、ホスト・コンピュータ１００は、トラステッド・モジュール１２０も含む。このトラステッド・モジュールは、通常の通信路１１０だけでなく、ＣＰＵ１０２、ハードディスク・ドライブ１０４、及び、ネットワーク・インタフェース・カード１０６の赤領域１１４への相互に独立で追加的な通信路１２２（１２２ａ、１２２ｂ、１２２ｃ）も備える。例示のため、トラステッド・モジュール１２０は、メモリ１０８とのそのような個別の追加的通信路１２２を有さない。

【０１４３】

トラステッド・モジュール１２０は、追加的通信路１２２ａ，ｂ，ｃのそれぞれを介して、ＣＰＵ１０２、ハードディスク・ドライブ１０４、及び、ネットワーク・インタフェース・カード１０６の赤領域１１４と通信することができる。また、トラステッド・モジュール１２０は、通常の通信路１１０を介して、ＣＰＵ１０２、ハードディスク・ドライブ１０４、ネットワーク・インタフェース・カード１０６の黒領域１１６、及び、メモリ１０８と通信することができる。さらに、トラステッド・モジュール１２０は、トラステッド・モジュールに格納されるポリシーの制御下において、トラステッド・モジュール１２０及び追加的通信路１２２を介して、ＣＰＵ１０２、ハードディスク・ドライブ１０４、及び、

ネットワーク・インタフェース・カード１０６の赤領域１１４の間で特定の情報をルーティングするための１００ＶＧスイッチングセンタとして機能することができる。また、トラステッド・モジュール１２０は、暗号鍵を作成し、追加的通信路１２２ａ，ｂ，ｃのそれぞれを介して、ＣＰＵ１０２、ハードディスク・ドライブ１０４、及び、ネットワーク・インタフェース・カード１０６の赤領域１１４へこれらの鍵を配布することもできる。

【０１４４】

図８は、トラステッド・モジュール１２０の物理的アーキテクチャを示している。第１のスイッチング・エンジン１２４は、追加的通信路１２２ａ，ｂ，ｃのそれぞれに個別に接続され、またトラステッド・モジュール１２０の内部通信路１２６にも接続されている。このスイッチング・エンジン１２４は、トラステッド・モジュール１２０内にロードされるポリシーの制御下にある。トラステッド・モジュール１２０のその他のコンポーネントは：

・トラステッド・モジュール１２０を管理し、トラステッド・モジュール１２０に対して一般目的の計算を行うコンピューティング・エンジン１２８と、；

・一時データを格納する揮発性メモリ１３０と、；

・長期データを格納する不揮発性メモリ１３２と、；

・暗号化および鍵生成などの特別な暗号機能を行う暗号化エンジン１３４と、；

・主に暗号化動作時に使用される乱数ソース１３６と、；

・トラステッド・モジュール１２０を通常の通信路１１０へ接続する第２のスイッチング・エンジン１３８と、；

・不正防止機構１４０と、；

からなり、これらはすべてトラステッド・モジュール１２０の内部通信路に接続されている。

【０１４５】

このトラステッド・モジュール１２０は、図１～図７を参照して上で詳述されたようなトラステッド装置またはトラステッド・モジュール２４を基礎にしている。

【０１４６】

暗号化鍵作成および配布について述べると、トラステッド・モジュール１２０は、乱数生成装置１３６、ハッシュ・アルゴリズム、及び、その他のアルゴリズムを用いて暗号鍵を生成するが、これらすべては、本質的にはセキュリティの当業者に周知である。トラステッド・モジュール１２０は、通常の通信路１１０ではなく、追加的通信路１２０ａ，ｂ，ｃのそれぞれを用いて、ＣＰＵ１０２、ハードディスク・ドライブ１０４、及び、ネットワーク・インタフェース・カード１０６の赤領域１１４へ選択された秘密鍵を配布する。鍵は、通常の通信路１１０上のプラットフォームの内部モジュール１０２、１０４、１０６、１２０間の通信のために使用される。外部データの大量暗号化または大量復号のため、トラステッドモジュール１２０がトラステッド・モジュール１２０の外部へ公開されてはならない長期間識別秘密を用いたＳＳＬのハンドシェーク・フェイズを完了した後、ＳＳＬプロトコルを利用して、（ネットワーク・インタフェース１０６またはＣＰＵ１０２により）他の一時的な鍵が用いられてもよい。ハードディスク・ドライブ１０４に格納されたデータの大量暗号化または大量復号のため、トラステッド・モジュール１２０の内部に一時的な他の鍵が生成され公開された後、トラステッド・モジュール１２０の外部へ公開されてはならない長期秘密を利用して、（ハードディスク・ドライブ１０４またはＣＰＵ１０２により）それら他の鍵が用いられてもよい。

【０１４７】

トラステッド・モジュール１２０は、暗号鍵の選択的な配布により、モジュール間通信のポリシー制御を実施している。トラステッド・モジュール１２０は、モジュールのペアの間に共用インフラストラクチャ１１０上のセキュアな通信を可能にする鍵の発行を拒否することにより、所与のモジュールの組の間の通信を禁止するポリシーを実施している。

【０１４８】

図９は、トラステッド・モジュール１２０により実施可能なウォッチドッグ（監視タイマー）機能及び、追加的通信路１２２に接続されるモジュール１０２，１０４，１０６に「ｐｉｎｇ」を行う処理を示している。トラステッド・モジュールはチャレンジ１４２を生成し、これをＣＰＵ１０２、ハードディスク・ドラ

イブ１０４、およびネットワーク・インタフェース・カード１０６の赤領域１１４へ、追加的通信路１２２ａ，ｂ，ｃのそれぞれを介して送信する。ＣＰＵ１０２、ハードディスク・ドライブ１０４、およびネットワーク・インタフェース・カード１０６のそれぞれは、モジュールがアクティブであるか否か、及び、好ましくはモジュールが適切に動作していることを伝えるため、追加的通信路１２２ａ，ｂ，ｃのそれぞれに応答１４４ａ，ｂ，ｃをそれぞれ応答する。トラステッド・モジュール１２０は、この応答１４４ａ，ｂ，ｃを記録し、図１～図７を参照して上述した完全性チャレンジへの応答の際、メトリックとしてこれらを用いる。

【０１４９】

図１０は、トラステッド・モジュール１２０がプラットフォーム内で暗号機能を有する唯一のモジュールである場合に、着信した外部セキュア・メッセージが処理される処理を示すものである。外部メッセージ１４６は、外部通信路１１２を用いてネットワーク・インタフェース・カード１０６の黒領域１１６により受信される。ネットワーク・インタフェース・カード１０６は、認証及び完全性検査を求める何らかのデータ及び要求とを含むプロトコル・データ・ユニット１４８（後で詳述）を、通常の通信路１１０を用いてトラステッド・モジュール１２０へ送る。トラステッド・モジュール１２０は、トラステッド・モジュール１２０の外部に公開されてはならないトラステッド・モジュール１２０内部の長期鍵を使用して、認証および完全性検査を実施し、、「ＯＫ」表示を含むプロトコル・データ・ユニット１５０を、追加的通信路１２２ｃを使用してネットワーク・インタフェース・カード１０６の赤領域１１４へ送信する。次に、ネットワーク・インタフェース・カード１０６は、復号を求める何らかのデータ及び要求を含むプロトコル・データ・ユニット１５２を、通常の通信路１１０を使用してトラステッド・モジュール１２０へ送信する。トラステッド・モジュール１２０は、トラステッド・モジュール１２０の内の一時的な鍵または長期鍵のいずれかを用いてこのデータを復号し、この復号されたデータを含むプロトコル・データ・ユニット１５４を、追加的通信路１２２ａを使用してＣＰＵ１０２へ送信する。その後、ＣＰＵは適切なアクションを実行する。

【０１５０】

図１１は、ＣＰＵ１０２が、トラステッド・モジュール１２０からのポリシー判定を要求する処理を示すものである。この処理は、たとえば、ポリシーが特定のデータが操作されること又はアプリケーションが実行されることを許可するか否かを、ＣＰＵ１０２が判定するときに用いられる。これについては、図１４～図２０を参照して後に詳しく説明される。ＣＰＵ１０２は、要求を含むプロトコル・データ・ユニット１５６を、通常の通信路１１０を使用して、トラステッド・モジュール１２０へ送信する。トラステッド・モジュール１２０は、トラステッド・モジュール１２０の内部に格納されたポリシーにしたがって要求１５６を処理する。トラステッド・モジュール１２０は、応答を含むプロトコルデータ・ユニット１５８を、追加的通信路１２２ａを使用してＣＰＵ１０２へ送信し、ＣＰＵ１０２がトラステッド・モジュール１２０から受信する権限を信用できるようにする。アクションに権限が与えられると、ＣＰＵ１０２は必要なアクションを実行する。与えられなければ、その処理を放棄する。

【０１５１】

図１２は、モジュール１０２，１０４，１０６間の保護された通信に対するポリシーの制御の例を示している。この例におけるすべての通信は、追加的通信路１２２を使用する。ネットワーク・インタフェース・カード１０６の赤領域１１４は、ハードディスク・ドライブ１０４宛てのプロトコル・データ・ユニット１６０を、追加的データ・パス１２２ｃ上のトラステッド・モジュール１２０へ送信する。ポリシーがこれを許可しない場合、トラステッド・モジュール１２０は、拒否を含むプロトコル・データ・ユニット１６２を、追加的データ・パス１２２ｃ上のネットワーク・インタフェース・カード１０６へ送信することによりその要求を拒否する。その後、ＣＰＵ１０２は、ハードディスク・ドライブ宛てのプロトコル・データ・ユニット１６４を送信することにより、ハードディスク・ドライブ１０４から機密にかかわるデータを要求するが、これは追加的データ・パス１２２ａ上のトラステッド・モジュール１２０へ送信される。トラステッド・モジュール１２０は、ポリシーがこれを許可することを検査する。これが許可された場合、トラステッド・モジュール１２０は、プロトコル・データ・ユニッ

ト１６４を、追加的データ・パス１２２ｂ上のハードディスク・ドライブ１０４へ中継する。ハードディスク・ドライブ１０４は、データをプロトコル・データ・ユニット１６６に含め、宛先をＣＰＵ１０２にして追加的データ・パス１２２ｂ上のトラステッド・モジュール１２０へ返送する。トラステッド・モジュール１２０は、ポリシーがこれを許可することを検査し、許可された場合、このプロトコル・データ・ユニット１６６を、追加的データ・パス１２２ａ上のＣＰＵ１０２へ中継する。

【０１５２】

図１３は、データが追加的通信路１２２上で受け渡される際のデータ・プロトコル・ユニット１７８のフォーマットを示すものである。このデータ・プロトコル・ユニット１７８は：

・プロトコル・データ・ユニットの種類を示す識別子フィールド１６８と、；

・プロトコル・データ・ユニットの長さを示す長さフィールド１７０と、；

・プロトコル・データ・ユニットの送信元を示す送信元フィールド１７２と、；

・プロトコル・データ・ユニットの宛先を示す宛先フィールド１７４などを有し、；

・多くの場合、データ・フィールド１７６を含む。

【０１５３】

すべてのフィールドが必ずしも必要なわけではない。たとえば、トラステッド・モジュール１２０のポリシーは、トラステッド・モジュールがトラステッド・モジュール１２０内からのものではない鍵プロトコル・データ・ユニットを中継することを禁止すると仮定すると、ＣＰＵ１０２、ハードディスク・ドライブ１０４、及び、ネットワーク・インタフェース・カード１０６は、鍵がトラステッド・モジュール１２０からのものであることを仮定することができる。したがって、送信元フィールド及び宛先フィールドは、鍵プロトコル・データ・ユニットに不必要となり、そのようなプロトコル・データ・ユニットは、暗黙で認証される。プロトコル・データ・ユニットの設計、構築および使用は、本質的には、通信の当業者にとって周知のものである。

【０１５４】

次に、本発明の特定の実施形態について、図１４～図２０を参照して説明する。図１４は、物理的なシステムを示しており、図２１と図７～図１３を参照して上述されたシステムの拡張である。図１４において、ディスプレイ１２１は、上述した追加的通信路のうち１つの手段１２２ｄにより、トラステッド・モジュール１２０に接続されている。これは、オペレーティング・システムを含む通常のソフトウェアからの破壊の恐れなく、トラステッド・モジュール１２０がディスプレイへ確実に書き込みできるようにしている。また、ホスト・コンピュータ１００は、内蔵型スマートカードリーダ１０３を有するキーボード１０１に接続されており、その両方は通常の通信路１１０に接続されている。スマートカードリーダ１０３に挿入されるスマートカードは、追加的なトラステッド・モジュールであると考えられ、これによって、トラステッド・モジュール１２０とセキュアに通信することが可能である。

【０１５５】

図１５は、トラステッド・モジュール１２０のコンポーネントの論理的な概略図を示しており、トラステッド・モジュール１２０内の使用許諾コード・コンポーネント２００及びその他の使用許諾データ・コンポーネント２０２からなる。前述のように、この使用許諾コード・コンポーネント２００は保護された環境内で動作し、好ましくはトラステッド・モジュール１２０自体の内部で動作するものであり、セキュア実行プログラム２０４、セキュア・ローダ２０６、セキュア鍵転送コード２０８、及び、クライアント・ライブラリ２１０を含んでいる。トラステッド・モジュール１２０上に格納されたライセンス関連データコンポーネント２０２は、トラステッド・モジュール１２０の秘密鍵２１２と、トラステッド・エンティティの公開鍵証明書２１４と、クリアリングハウスまたは開発者の公開鍵証明書２１６と、使用許諾ログ２１８と、公開鍵証明書２１４を有するトラステッド・エンティティの秘密鍵を用いて署名されるライセンス関連コード２００のハッシュされたバージョン２２０とを含む。

【０１５６】

図１６は、クライアント・コンピュータ１００内の保護されたソフトウェアまたはデータ２２２の構造を示している。クライアント・コンピュータ１００上の

デジタル・データ２２４は、それぞれソフトウェア実行プログラム２２６に関連づけられており、これらのソフトウェア実行プログラムの内部には、トラステッド・モジュール１２０の公開鍵２２８が格納されている。この構造２３０は、クリアリングハウスまたは開発者の秘密鍵を用いて署名された構造２３０の、ハッシュされたバージョン２３２とともに格納されている。それぞれの保護されたソフトウェアまたはデータの一部に対して、ユニット２２２に類似した構造が得られる。

【０１５７】

図１７は、トラステッド・モジュール１２０内でセキュア・ローダ２０６が動作していないかもしれない一般的な場合について、クライアント・プラットフォームへソフトウェアまたは他のデータをロードまたはアップグレードするためのフローチャートを示している。

【０１５８】

インストールされるデータは、ハッシュされ、送信者の秘密鍵を用いて署名され、送信者によって、これがそのデータ自体に添付される。

【０１５９】

ステップ２３４において、オペレーティング・システムは、データがインストールされるセキュア・ローダ２０６へ、データ及び署名されハッシュされたバージョンと共に要求を送信する。ステップ２３６において、セキュア・ローダ２０６は、この要求を受信し、ステップ２３８において、セキュア・ローダ２０６が、送信者に対応する公開鍵証明書を用いてこのメッセージの署名を検査し、これによって送信者の認証を検査する。

【０１６０】

認証が失敗すると、ステップ２４０において、セキュア・ローダ２０６は、オペレーティング・システムへエラー・メッセージを送信する。ステップ２４２において、オペレーティング・システムは、このエラーメッセージを受信し、ステップ２４４において、適切なメッセージを表示する。

【０１６１】

ステップ２３８において、認証が成功すると、ステップ２４６において、セキ

ュア・ローダ２０６は、トラステッド・モジュール１２０内で使用可能な暗号化機能によって、このメッセージのハッシュを計算し、ステップ２４８において、これをステップ２３６でデータと関連して受信されたメッセージ・ハッシュと比較する。これにより、メッセージの完全性が検査される。

【０１６２】

このハッシュが同じでない場合、これはデータが改竄されたこと、及び、そのデータがインストールされてはならないことを示している。この場合、ステップ２５０において、セキュア・ローダ２０６がＯＳへエラー・メッセージを送信し、次にＯＳが上述されたステップ２４２および２４４を実施する。

【０１６３】

ステップ２４８においてこのハッシュが同一であると判定された場合、ステップ２５２において、トラステッド・モジュール１２０がインストールのログ記録を行い、ステップ２５４において、セキュア・ローダ２０６は、データが普通にインストールできることをＯＳに示し、次にステップ２５６においてそれが実施される。

【０１６４】

検査の他の形態（典型的にはライセンス検査）が、追加または代替として使用される場合、これらは、図１７を参照して説明される方法において、ステップ２５０とステップ２５２の間に含めることが可能である。

【０１６５】

図１８は、ＯＳがセキュア実行プログラム２０４と通信するライセンス検査のモデルを使用した使用許諾のためのフローチャートを示しており、データの一部に関連づけられたソフトウェア実行プログラム２２６は、そのデータの保護に使用されるライセンスモデルを選択するためのオプションを有する。これもまた、使用許諾ソフトウェアがトラステッド・モジュール１２０内に取り付けられる必要がない一般的な場合に対するものである。この処理は、次のようなものである。

【０１６６】

ユーザが何らかのデジタル・データを動作させたいと考えた場合、ステップ２

５８において、オペレーティング・システムにより要求が送信され、ステップ２６０において、セキュア実行プログラム２０４によりこれが受信される。ステップ２６２において、セキュア実行プログラム２０６は乱数（ノンス）を生成し、ステップ２６４において、このノンスをトラステッド・モジュール１２０の秘密鍵２１２を用いて署名されたアプリケーションへの参照（たとえばその題名）と共に送信することにより、そのデータの一部に対応するソフトウェア実行プログラム２２６へチャレンジ／レスポンスを発行する。

【０１６７】

ステップ２６６において、これがソフトウェア実行プログラム２２６により受信されると、ステップ２６８において、ソフトウェア実行プログラム２２６はトラステッド・モジュール１２０の公開鍵２２８を用いてセキュア実行プログラムのチャレンジを検証および認証する。エラーがあった場合、または、ソフトウェア実行プログラム２２６がこの特定のマシン上でデータが実行されることを望まない場合、ステップ２７０において、エラー・メッセージが送信され、ステップ２７２において、セキュア実行プログラム２０４によりオペレーティング・システムへ中継される。ステップ２７４においてエラー・メッセージが受信されると、ステップ２７６においてオペレーティング・システムが適切なエラー・メッセージを表示し、そのデータは実行されない。

【０１６８】

ステップ２６８においてエラーがなければ、ステップ２７８において、ソフトウェア実行プログラム２２６は、ノンス、データへの参照、及び、任意でライセンスモデルを含むメッセージをセキュア実行プログラム２０４へ返す。ノンスは、再生攻撃に対する保護を与えるために含まれる。

【０１６９】

ステップ２８０においてメッセージの受信が完了すると、ステップ２８２において、セキュア実行プログラム２０４は、ソフトウェア実行プログラムにより指定されたライセンスモデルにしたがって適切なライセンス検査を実行する。これは、鍵を使用してデータをロック解除することを含むことも可能である。これらのライセンスモデルのさらなる詳細は後述される。そのデータと関連づけられた

ソフトウェア実行プログラムがない場合、セキュア実行プログラムは、管理者によってその内部に事前に設定されたデフォルトのライセンスモデルに対応するライセンス検査を実行する。有効なライセンスが存在する場合、ステップ２８４において、セキュア実行プログラム２０４はトラステッド・モジュール１２０にトランザクションの計測記録を行うよう依頼し（ステップ２８６，２８８）、ステップ２９０において、そのデータを実行する許可をオペレーティング・システムへ送信する。ステップ２９２において受信があると、ステップ２９４において、オペレーティングシステムがデータを実行する。ステップ２８２のライセンス検査の後、有効なライセンスがなければ、ステップ２９６において、セキュア実行プログラム２０４は、オペレーティング・システムに、これを適切にエンドユーザへ通知するよう依頼し（ステップ２７４および２７６）、データは実行されない。

【０１７０】

図１９は、ＯＳがセキュア実行プログラム２０４ではなくソフトウェア実行プログラム２２６と通信する場合のライセンス検査のモデルを使用した使用許諾のフローチャートである。このモデルも、使用許諾ソフトウェアがトラステッド・モジュール１２０内に取り付けられる必要がない一般的な場合に対するものである。

【０１７１】

ユーザが何らかのデータを実行を希望する場合、ステップ２９８において、ＯＳはそのデータと関連づけられたソフトウェア実行プログラム２２６へメッセージを送信し、ステップ３００において、これが受信される。ステップ３０２において、ソフトウェア実行プログラム２２６は乱数（ノンス）を生成し、ステップ３０４において、このノンスをデータへの参照とともに送信することにより、トラステッド・モジュール１２０内のセキュア実行プログラム２０４へチャレンジ／レスポンスを発行する。加えて、スマートカードＩＤがクライアントマシンへのログインに用いられ、ホットデスキングが用いられるべきライセンスモデルである場合には、スマートカードＩＤも送信される。

【０１７２】

ステップ３０６においてメッセージの受信があると、ステップ３０８において、セキュア実行プログラム２０４はデータに対して適切なライセンス検査を行う。有効なライセンスが存在しない場合、ステップ３１０において、セキュア実行プログラム２０４はエラー・メッセージを返し、このエラーメッセージからソフトウェア実行プログラムが使用許諾に伴う正確な問題の種類を判定し、これを適切にＯＳへ報告することを可能している（ステップ３１２，３１４，３１６）。

【０１７３】

有効なライセンスがある場合、ステップ３１８において、セキュア実行プログラム２０４は、ノンスとデータへの参照とを含みトラステッド・モジュール１２０の秘密鍵２１２を使用して署名され暗号化されたメッセージを返す。ノンスは、再生攻撃に対して保護を与えるために含まれる。

【０１７４】

ステップ３２０におけるメッセージの受信後、ステップ３２２において、ソフトウェア実行プログラム２２６は、セキュア実行プログラムの返答が正しいか否かを、トラステッド・モジュール１２０の公開鍵証明書２２８を使用して検証する。検証が正しい場合、ステップ３２４において、ソフトウェア実行プログラム２２６はトラステッド・モジュール１２０にログの作成を依頼し（ステップ３２６，３２８）、ステップ３３０においてデータを実行（ステップ３３２，３３４）するための呼び出しをＯＳに渡す。一方、検証が正しくない場合、ステップ３３６において、ソフトウェア実行プログラム２２６はＯＳへエラー・メッセージを送信し、ＯＳはエラー・メッセージを適宜表示する（ステップ３１４，３１６）。

【０１７５】

デジタル・データを実行するための許可についての検査を強制的に実行する好適な機構において、トラステッド・モジュール１２０は、本発明を実施するためのハードウェア、及び／または、ソフトウェアを含んでいる。典型的には、トラステッド・モジュール１２０は、アプリケーションとＯＳとの間のブリッジとして動作する。ＯＳは、トラステッド・モジュールからの要求を除き、好ましくは通常のアプリケーション及びＯＳでないソフトウェアへアクセスできないトラス

テッド・モジュール１２０とＣＰＵ１０２の間の通信路１２２を介して与えられる、アプリケーションをロードまたは実行することのすべての要求を無視することが好ましい。ホスト・コンピュータ上で動作する処理は、次のようになる。まず、好ましくはそのデータと関連づけられたソフトウェア実行プログラム２２６を介して、及び、通常エンド・ユーザによる何らかのアクションに応答して、アプリケーションまたは他のデータを実行するための、トラステッド・モジュール１２０への最初の要求がある。ソフトウェア実行プログラム２２６は、データがインストールされた、または、インストールされるべきトラステッド・モジュール１２０の公開鍵証明書２２８を保持することになる。トラステッド・モジュール１２０内のセキュア実行プログラム２０４は、上述のように適切なライセンス検査を実行する。この検査の結果、データの実行が適切である場合、セキュア実行プログラム２０４は、この情報を、通常のアプリケーションとＯＳ以外のソフトウェアはアクセスできないことが好ましいＣＰＵ１０２への通信路１２２を介して、ＯＳへ情報を送信する。その後、ＯＳは、アプリケーションまたはデータを実行するためにホスト上で処理を開始する。データのインストールが適切であることを示すためにセキュア・ローダがＯＳと通信するとき、または、ロック解除鍵を転送するために鍵転送コードがＯＳと通信するときには、同様の処理が行われる。

【０１７６】

図２０は、上述のようなライセンス検査のモデルを使用した使用許諾のためのフローチャートを示すものであり、この例では、使用許諾を行うソフトウェアはトラステッド・モジュール１２０内に格納されていて、トラステッド・モジュール１２０はアプリケーションとＯＳとの間のブリッジとして動作する。この処理は、セキュア実行プログラム２０４がトラステッド・モジュール１２０自身内に存在すること、及び、セキュア実行プログラムがＯＳと通信しているときには、セキュア実行プログラムはトラステッド・モジュール１２０からＣＰＵ１０２への通信路１２２（専用であることが好ましい）を使用するという点を除いて、図１９で与えられるものと同様である。

【０１７７】

本発明が使用されることが可能な多くの異なる方法が存在する。次にそれらのうち６つの詳細を説明する。

【０１７８】

例Ａ：

第１の例は、アプリケーションをハードウェアに結合することにより、不正耐性ハードウェアを汎用ドングルとして使用する例である。この節におけるこの例とその他の例との大きな違いは、第１に、コードが実際に実行されているときにライセンス保護が実行されること、第２に、この方法は、保護機能を実行しているパーティがソース・コードを利用可能であるアプリケーションの保護に適していることである。

【０１７９】

ソフトウェアがプラットフォーム内へロードされる（さらに任意で、それが動作するであろう不正耐性ハードウェアへロードされる）。このソフトウェアは、セキュア・ローダを用いて完全性検査が行われる。トラステッド・モジュール内の秘密の存在を検査するため、または、そのトラステッド・モジュールの識別及び存在を検査するため、ＡＰＩコールがトラステッド・モジュールへ用いられる。さらに、トラステッド・モジュールは、コードの一部を実行されることが可能である。トラステッド・モジュールの秘密暗号鍵及び標準認証プロトコルを使用することにより、トラステッド・モジュールの強力な認証が可能である。

【０１８０】

さらに、次のオプションが存在する。：

・ＯＳではなくトラステッド・モジュールへＡＰＩコールが実行される（前述のように）。；

・トラステッド・モジュールは、コードの一部を実行されることが可能である。これは、いくつかの方法で成されることが可能であり、そのうちいくつかは既に説明した。；

・コードの一部は、不正耐性ハードウェア（内部トラステッド・モジュールまたはスマートカードなど）内への移動のためにマークされることが可能であり、その場合、コードは暗号化された形態で格納され、呼び出し（コール）はコード内

の他の場所にあるこの機能へ成される。；

・同様に、スマートカードなどの可搬式トラステッド・モジュールは、コードの一部を実行するようにされることが可能である。

【０１８１】

ハードウェア・ドングルへのＡＰＩコールの同様の使用よりも、この方法の使用の方が、このアプローチと通常関連する欠点の多くに対抗する。

【０１８２】

第一に、ハードウェア・ドングルへのＡＰＩコール用いる従来のソフトウェア保護は、デバッガまたは逆アセンブラーによるソフトウェア・ロックの変更を受けやすく（たとえば、プロセッサとマザーボードとの間の通信に介入することなどにより）、したがって、鍵へのコールを取り除くためのコードを変更している。変更されたコードのコピーが生成され、ホスト・マシン上と他のマシン上との両方において、自由に動作する。このようなことは、本方法では、次のようなことによって対抗することができる：

・トラステッド・モジュール自身の内部では、コードの一部が動作している。

・関連するライセンス検査コードが確実にソフトウェアと共にロードされることを保証し、ライセンス検査がバイパスされていることを防止する、プラットフォーム上の完全性検査及び関連するソフトウェア。

【０１８３】

第二に、ハードウェア上で実行される処理の欠けている機能のいくらかを埋めるため、記録と再生（または他の技術）が用いられるかもしれないという危険性が現在存在する。この危険性は、本方法においては、ソフトウェア及びライセンス検査コードの完全性検査により対抗できる。

【０１８４】

第三に、ライセンスがマシンに縛られる必要がないこと、及び、支払いモデルの選択枝がより大きくなることの両方により、ライセンスモデルに非常に大きな柔軟性がある。トラステッド・モジュールは、特定のアプリケーションにのみ特化されるものではない汎用ドングルを提供し、さらに、ライセンス情報格納の能力をより大きくし、より優れた計測を提供する。

【０１８５】

最後に、開発者の労力に関連する利点もある。ソフトウェアへＡＰＩコールを追加することの利点は、ソフトウェアが特定のマシン用にカスタマイズされることであり、したがって、実行コードまたはソース・コードが明確（クリア）に獲得されたとしても別のマシン上での直接の利点はない。しかしながら、開発者の一部にかなりの労力を要求する可能性もある。唯一の違いが異なるトラステッド・モジュールＩＤであることにより、コードの完全性検査による保護があれば、開発者は非常にわずかな労力で実質的な保護が得られる。ここでも、トラステッド・モジュール自体の内部にあるコードの動作する部分は、コードの個別カスタマイズを必要としない。

【０１８６】

この例において：

・開発者は、以下の中から任意の組み合わせを行うことができる

・ＡＰＩコールをソフトウェア、及び／又は、そのソフトウェアと関連するソフトウェア実行プログラム内へ挿入する。これらは、次のものを検査する：

・不正耐性装置内の秘密の存在（たとえば、開発者がスマートカード・ドングルを作成し、それらをエンドユーザへ出荷した場合）または；

・エンド・ユーザのマシン内の（汎用ドングルとして使用している）不正耐性装置の識別および存在。；

ソフトウェア実行プログラムは、通常、動作時に検査のみを行う。；コード内のさらなるＡＰＩコールは、要求があれば、コードの実行中さまざまなステージにおいて実行されることが可能である。これは、ソフトウェアにとって一般的な方法で行われ（すなわち、顧客のそれぞれは同じバージョンを受け取る）、正しいトラステッド・モジュールＩＤなどのカスタマイズされた詳細は、後に後述の登録ステージで追加されることができる。；

・トラステッド・モジュールまたは他のトラステッド装置内の秘密の存在を検査することによる使用許諾方法が使用されるべきことをコンピュータ・プラットフォーム内のセキュア実行プログラムへ通知する情報と共に、そのデータと関連するソフトウェア実行プログラムへ秘密を挿入する。たとえば、ｌｉｃｅｎｓｉ

ｎｇ＿ｍｅｔｈｏｄ（秘密，ｓｃ，ｋ，ｗ）またはｌｉｃｅｎｓｉｎｇ＿ｍｅｔｈｏｄ（秘密，ｔｃ，ｋ，ｗ）は、現在のスマートカードまたはマシン内部トラステッド・コンポーネント内に秘密ｋが格納されていることが判明したならば、ｗにより参照されるソフトウェアはマシン上で実行のみが許可されなければならないことを示している。セキュア実行プログラムは、この検査の実行を可能にし、検査が成功しない限りソフトウェアｗの実行を許可しないということがあらかじめ記憶されたプロトコルを有する。

【０１８７】

・ユーザは開発者に登録する。初期化処理の一部として、使用許諾システム内の通信パーティ間の認証は、それらの間で送信されるメッセージの機密性に対するセッションキーの交換の前に（または使用されているプロトコルによっては同時に）行われる（この処理のさらなる詳細については例Ｂを参照）。不正耐性コンポーネントへ、その開発者に対応する公開鍵証明書が送信される。支払いに応答して、

（１）ユーザは、コードにより検査された開発者の秘密を含む（メモリまたはハードコーディングにより）可搬式ハードウェア耐性装置（スマートカードなど）と共に一般的にカスタマイズされたソフトウェアが与えられる、あるいは、ユーザの不正防止装置へ鍵が転送される（例えば、この鍵はソフトウェアの復号のためのロック解除鍵ではないという点を除いて下記の例Ｂで詳述されるものと同様の方法により転送される）。（２）ソフトウェアにユーザのマシンＩＤが挿入されて（ＡＰＩコールがその特定のマシンＩＤを検査するため）ユーザへ出荷される。

【０１８８】

・アプリケーションとトラステッド・モジュールの間の対話を制御するために、開発者は、２つの追加コンポーネント、すなわちソフトウェア実行プログラムおよびクライアント・ライブラリを顧客へ出荷する必要がある。このクライアント・ライブラリは、アプリケーションがソフトウェア実行プログラムと通信するために呼び出す上位インタフェース・サブルーチンの集合である。

【０１８９】

・前述の２つのステージにおいて説明したソフトウェアおよびコードは、メッセージに添付される送信側の秘密鍵により署名された、メッセージのハッシュされたバージョンを用いて署名され、受信側がメッセージの完全性を検査できるようにしている。より明確に述べれば、開発者はコードＭをハッシュし、開発者の秘密鍵（Ｓｐｒｋ）を用いてこれに署名し、署名Σsprk（ｈ（Ｍ））を生成する。その後、開発者は、この署名をメッセージＭと共に送信する。

【０１９０】

・その後、セキュア・ローダは、開発者の公開鍵を用いてこの署名を検査し、メッセージ・ハッシュを取り出す。これにより、その送信者が、その署名に用いられた公開鍵の持ち主であることが保証される。メッセージとメッセージ・ハッシュを得ると、セキュア・ローダはこのメッセージのハッシュを計算し、これを、復号されたメッセージ・ハッシュと比較する。これは、メッセージの完全性を検査している。さらに、この完全性検査機構は、チャレンジ／レスポンス、またはハッシュ内に通信の履歴を導入などの、何らかの標準的な機構により再生攻撃を防止する。

【０１９１】

完全性検査が作動すると、セキュア・ローダはソフトウェアをインストールする。これにより、変更の施されたソフトウェア（ＡＰＩコールを持たないソフトウェアなど）が動作できないこと、ウィルスが侵入していないことなどが保証される。また、このソフトウェアは、インストール時にプラットフォーム内のトラステッド・モジュールの存在を検査するように修正されることも可能である。

【０１９２】

・ユーザがソフトウェアを動作させようとするときには、ソフトウェア実行プログラムが全体の制御権を奪取し、実行の開始時に初期検査を行う。これらの検査が満足されると、ソフトウェア実行プログラムはソフトウェアの動作を許可する。もし追加的なＡＰＩコールがこのソフトウェアに組み込まれている場合には、これらは、動作中のさまざまな時点において、トラステッド・モジュールに対して行われる。

【０１９３】

・そのような検査が行われると同時に、ソフトウェアが問題無く実行された場合、トラステッド・モジュール内に記録が作成される。支払いモデルによっては、使用報告がクリアリングハウスまたは登録団体へ送信されることもできる。特定回数のソフトウェアの実行に対する支払いは、たとえばスマートカードを利用して、容易にモデル化することが可能である。

【０１９４】

例Ｂ：

第２の例は、データの一部またはすべてを暗号化することによって、トラステッド・モジュールを汎用ドングルとして使用する。この例においても、セキュア・ローダによる実行されるデータの完全性検査と、セキュア・ローダ、セキュア実行プログラム、ソフトウェア実行プログラム、およびセキュア転送コードの完全性検査とが存在する。トラステッド・モジュールのトラステッド識別（秘密暗号鍵）は、強力な認証を実施するために使用される。任意であるが、アプリケーションは、トラステッド・モジュールまたはスマートカード内で動作することが可能である。このような使用許諾システムの一般的な利点は、ライセンス管理システムの柔軟性を、ドングルの短所を有することなく、高度のハードウェア・セキュリティと組み合わせることが可能になるという点にある。

【０１９５】

特に、現在の使用許諾システムに伴う問題は、以下のようにして解消される。

：

・ライセンス検査のバイパス（回避）は、プラットフォーム上の完全性検査により解消される。この検査は、トラステッド装置が除去または改ざんされた場合、または使用許諾ソフトウェアが変更された場合に失敗となる。

・現在の一般的なデータ保護の方法の欠点は、データが実行される時点まで保護されていても、いったん実行可能がロック解除されるまたは使用可能となると、データが自由に複製及び使用される可能性がある。

・依然としてデータが複製される可能性はあるが、本発明を組み入れたいずれの他のセキュアプラットフォームにおいても、必要なライセンスなしにデータが実行されることはない。

・ドングルは、特定のアプリケーションのためにあつらえられたものではなく汎用的なものである。

・支払いまたはライセンスモデルに柔軟性がある（異なる種類の使用許諾の組み合わせを可能とすることを含む）。

・ハードウェア装置内に汎用システム鍵を持つことを避けることができ、開発者及びハードウェアの秘密鍵を秘密に保つことができる点において、Ｗａｖｅ　Ｓｙｓｔｅｍｓ　ＷａｖｅＭｅｔｅｒなどの汎用ドングルに対して改善されている。このことは、第三者が信用できない場合に特に重要である。なぜなら、ロック解除鍵を知らないであろうクリアリングハウスまたは誰か他の者が、保護されたデータを使用することを可能にするからである。このことは、クリアリングハウスによってこの鍵を知られることになる現在のシステムを改善している。

・トラステッド・モジュール間のライセンスの自動転送により、鍵管理の問題が回避される。

・各開発者は、一般コンテンツ保護または専用コンテンツ保護のいずれも選択できる。Ｋ（またはＫ′）は、必要であれば顧客ごとに異っていてもよい。これにより、開発者に大きな柔軟性を与え、さらなるセキュリティに対する労力のバランスをとることを可能とする。より一般的に述べるならば、各種のライセンスモデル（たとえば、例Ａ、ＢまたはＣに対応するモデル）は、各顧客に出荷されるデータに基づいて同一に使用される、または、個別にカスタマイズされることが可能である（したがって、他のマシンでは使用できない）。これらの方法の組み合わせを、同一のプラットフォームに使用することも可能である。したがって、開発者には、自分の使用したいデータ保護の種類について選択枝が与えられる。開発者は、ロック解除鍵または一般保護の種類を、顧客毎に異なるように作成することもできるし、または同一にしてもよい。クライアント・プラットフォームは、この選択について知らされる必要はない。

【０１９６】

この例において：

・汎用セキュア実行プログラム、セキュア・ローダ、およびセキュア鍵転送コードは、いずれのトラステッド・コンピュータ・プラットフォーム内にも含まれる

。このコードは、完全性検査が失敗するとロードされず、その場合、全体的なクライアント・プラットフォーム完全性検査は、本明細書で前述したように失敗しなければならない。

【０１９７】

・エンドユーザＡは、（支払いモデルにしたがって）開発者、サーバまたはクリアリングハウスＣに自分のマシン（トラステッド装置ＩＤ）を登録し、何らかのデータを受信するために、適切な支払いを行うように構成する。代替として、このハードウェア装置は、データ購入を記録し、これを後日Ｃへ報告するように構成されることもできる。

【０１９８】

・初期化処理の一部として、使用許諾システム内の通信パーティ間の認証は、メッセージの機密を保つため、セッション・キーの交換の前に（または使用されるプロトコルによっては、交換と同時に）行われる。

【０１９９】

・認証：Ｃからクライアントの不正防止装置へ認証がある。これは、Ａのトラステッド・モジュールからノンスを有するＣへのチャレンジを含む標準プロトコルを用いて行われ（再生攻撃に対する保護を提供するため）、Ｃは、Ｃの秘密コード署名鍵を用いてデジタル的に署名された、ノンスを含むメッセージを用いて応答する。任意で、Ａの不正防止装置からＣへの認証がある。Ｃの秘密コード署名鍵に対応する公開鍵Ｗを与える公開鍵証明書が、エンドユーザのトラステッド・コンポーネントへ転送される（場合によっては、（たとえばアップグレードなど）、これはすでにトラステッドモジュール内に存在する）。これにより、マシンは、ベンダの識別、及び、後に受信するアップグレード・データの完全性を検査できるようになる。ユーザ・ベースのライセンスモデルが利用される場合、この転送は可搬式トラステッド装置へ（たとえばスマートカードへ）行われる。また、Ｃには、Ａの不正防止装置内の秘密鍵Ｐに対応する公開鍵も与えられる。この場合、ＡからＣへの何らかの種類の認証が必要とされ、対称暗号を用いる場合には、鍵は非対称鍵のペアを用いて設定する（下記参照）。同様の方法において、開発者とクリアリングハウスとの間の公開鍵証明書は、これらが個別のパーテ

ィである場合、最初に交換され、適切な認証が行われる必要がある。上述のように、同一のプロトコルを用いることも可能である。

【０２００】

・対称鍵Ｋを使用して暗号化されたデータは、Ｃの秘密コード署名鍵の下で（たとえばマイクロソフト社のＡｕｔｈｅｎｔｉｃｏｄｅを使用して）署名され、ＣによりＡのマシンのエンドユーザへ送信される。必要であれば、Ｋは顧客毎に異なっていてもよい。保護される必要があるのはロック解除鍵であるため、このデータは、任意の手ごろな手段（例えば、インターネットまたは衛星放送など）によりエンドユーザへ転送される。暗号化に要する時間は、このステージでは問題とされないため、秘密鍵Ｋ’の代わりにオプション用いられる。

【０２０１】

機密性：個別の開発者とクリアリングハウスとが存在する場合、対称鍵のペアを準備する開発者とクリアリングハウスの間には、プロトコルが用いられる。このプロトコルは、例えばデータの支払い及び使用についての通信などの、これらの間の通信を暗号化するのに用いることができる。これらの手段により、いずれのパーティも、他のパーティの秘密鍵を知ることがないようにされる。保護されるべき各メッセージのコンテンツは、ランダムに生成されたＤＥＳキーを用いて暗号化され、意図する受信者の公開鍵を用いてＲＳＡ暗号化された対称鍵と共に転送される。この場合も、他のパーティに対応する公開鍵証明書は、各パーティへ最初にインストールされる必要がある。真正性及び完全性の検査が追加される場合、それぞれのメッセージに対するプロトコルは次のようなものになる：送信者は、ＤＥＳキーを作成する（乱数発生器を用いて鍵を生成し、これらの鍵が一回しか用いられないことを保証する）。その後、送信者はそのＤＥＳキーを用いてデータＤを暗号化し、また、受信者のＲＳＡ公開鍵を用いてそのＤＥＳキーを暗号化する。次に、送信者は、認証と完全性を提供するため、この情報のすべてのハッシュを署名し、暗号化されたデータと暗号化されたＤＥＳキーをこの署名と共に送信する。なお、機密のデータＤは、ＤＥＳキーを用いて暗号化された状態で格納される。そして、受信者のみが、ＤＥＳ暗号鍵を復号するためのＲＳＡ秘密鍵を有するはずであり、データＤを復号するためにこれを使用する。

【０２０２】

・ＡとＣの間のすべての通信は、前のステージで説明したように、ＤＥＳセッション・キーを用いて暗号化される。

【０２０３】

・さらに、Ｋに対応する対称ロック解除鍵（もしくは、Ｋ′に対応する公開鍵）は、Ａの公開鍵を用いて暗号化され、Ｃの秘密コード署名鍵を用いて署名され、データを動作可能にするため、エンドユーザの不正防止コンポーネントへ送信される。

【０２０４】

・エンドユーザのプラットフォームにより受信されると、Ｗを用いて署名を検査し、データが期待するソースから得られたものであるか否かを検証することにより、セキュア・ローダによってそのデータの完全性検査が実施される。

【０２０５】

・完全性検査が成功すると、データはプラットフォーム上にインストールされ、トラステッド・コンポーネントはこのイベントを記録する。成功しない場合、エラー・メッセージが生成され、データはロードされない。

【０２０６】

・エンドユーザのＰＣと関連する不正防止装置は、この情報を利用してロック解除鍵を取得できる唯一のものである。鍵転送コードは、完全性と認証についてメッセージを検査し、ロック解除鍵を復号し、これをそのデータに関連するトラステッド・モジュール上に格納する。

【０２０７】

ユーザがデータを動作させたい場合、セキュア実行プログラムが、ロック解除キーを用いてそのデータを復号し、データを動作可能にする。ロック解除鍵の実際の機能は、多様である：たとえば、プログラムの一部は、起動時またはインストール時に復号されることも可能であり、鍵自体は、不正防止コンポーネントの識別を入力として用いることにより形成されることも可能である。

【０２０８】

・不正防止コンポーネントは、データの使用をローカルに、及び、信用できる

方法で監視するためのログを保持する。

【０２０９】

例Ｃ：

第三の例は、トラステッドモジュールの識別と関連するデータベース情報またはプロファイル情報を参照することによる使用許諾の例である。

【０２１０】

この例は、登録および支払いに応じてライセンス・データベース・エントリを更新することを含んでいる。このアプローチを用いる主要なオプションが２つある。

例Ｃ１：

第一の例は、そのデータのロック解除鍵を得るため、セキュア実行プログラムが、データベース内において、トラステッド・モジュールＩＤエントリを検査する例である。このデータは、鍵を用いた暗号化または部分暗号化により保護されるため、侵害される恐れなく自由に配布されることが可能である。；

例Ｃ２：

第二の例は、そのデータの一部を動作させる許可を得るため、セキュア実行プログラムまたはソフトウェア実行プログラムが、データベース内において、トラステッド・モジュールＩＤエントリを検査することである。トラステッド・モジュールのＩＤに対応するエントリは、特定のアプリケーションを実行するための許可を示すように更新され、このデータベース上の許可が検査されると、セキュア実行プログラムまたはソフトウェア実行プログラムは、データを実行可能になる。この場合、データは汎用的なもので保護されておらず、自由に複製されることが可能であるが、当然ながら、必須の許可が存在しない場合、この種類のプラットフォーム上で実行させることはできない。セキュア実行プログラムがデータの実行を可能にすると、トラステッド・モジュールはそのログを更新する。ソフトウェア実行プログラムを用いて検査を実施する場合、実行されるべきアプリケーションと関連するソフトウェア実行プログラムがトラステッド・モジュールを呼び出し、トラステッド・モジュールがライセンス検査を実施し、その後、この検査が成功したならば、ソフトウェア実行プログラムがこの呼び出しをＯＳへ渡

し、アプリケーションが実行される。

【０２１１】

このアプローチの利点は、以下の通りである：

１）ドングルの欠点を有することなく、ライセンス管理システムの柔軟性を高度のハードウェア・セキュリティに組み合わせることが可能である。

２）そのような方法を使用する主な動機は、鍵管理上の理由による。特に、代わりのパスワードを発行するのは、面倒である。この方法では、更新されなければならないのはデータベースのみであるため、この問題を解消する。

３）ディレクトリ・システムがすでに存在する場合、この使用許諾方法は、セキュアなライセンス検査を提供するのに特別な大きな投資を必要としないため、自然な選択となる。

４）上記の例Ｃ１は、例Ｂと比較されるように、クライアント・マシンへロック解除鍵を与える別の方法に対応する。これは、２つの理由で好ましい。第一に、ディレクトリ・システムが適所に存在するかもしれないため、特定の企業にとっては好ましい解決策である。第二に、例Ｂでは可能でなかったが、この方法は、ロック解除鍵の非永久的な記憶を可能にし、浮動ライセンスを可能にしていることにある。

【０２１２】

現在使用可能なライセンス処理は、指紋情報をライセンスデータベースと照合し、その指紋に対応する有効なライセンスが存在するか否かを確かめることである。アプリケーションは、この情報に応じて実行可能または実行不可となる。ただし、この方法は、以下の理由により実際には使用されない。：

・ライセンス検査コードは、現時点では、容易にバイパス（回避）され得る。

・データベースの生成及びデータベースの更新に伴うオーバーヘッドが存在する。

・ＩＤを偽って、別のマシンまたは別のユーザにライセンスされている情報へのアクセス権を入手することが可能である。

【０２１３】

しかしながら、関連するライセンス検査コードの完全性検査と共に不正防止装

置を用いることにより、類似の方法を用いることが可能である。

【０２１４】

この方法は、既存の処理と関連した問題を克服するものである。：

・ディレクトリ構造は、使用許諾を行うことができるように拡張されることができる（ライセンス管理を参照）。－そのような構造は既に存在しており、追加的機能との統合を可能にしている。ライセンス・データベースは、トラステッド・コンポーネント内に格納されたローカル・レコード、サーバ内に格納されたレコード（必要な時に参照されローカルに格納される）、または、一元管理されるディレクトリ・サービスなどの形態を取ることが可能であり、アクセスについての適切な情報が格納されている。実際には、これらの組み合わせを用いることも可能である。一般的にＸ．５００として知られるディレクトリ標準は、サービス・プロバイダ、官庁、および民間の組織などに属するコンピュータ・システムを相互接続する多目的分散ディレクトリ・サービスのための基盤を提供する。そのようなディレクトリを修正し、コンピュータ・ネットワークのユーザが、ある人のユーザＩＤまたはマシンＩＤを検索したときに、その個人またはマシンにライセンスされたアプリケーションの詳細を含む情報を返せるようにすることは簡単であろう。

【０２１５】

・完全性検査は、ライセンス検査コードに行なわれ、またデータにも行われる。コンピュータ・プラットフォーム上の関連するソフトウェアは、ユーザまたはマシンがそのアプリケーションを実行するための許可を有しているか否かを検査し、これを適切に許可または禁止する。代替として、データがたとえば暗号化により保護されている場合には、異なるデータ・アクセス・キーがディレクトリ内に格納され、関連したソフトエアを介して、本方法で得られる方法によりそれらへアクセスすることが可能である。

【０２１６】

・より良い認証は、ディレクトリ／プロファイルのアプローチを可能とする。トラステッド・モジュール内のトラステッドＩＤ（ユーザＩＤの場合、生物測定学と組み合わせることも可能）は、強力な認証を可能にし、偽称の防止に役立つ

。（マシンまたはユーザ識別の信用度を高くすることにより、例えば別のユーザの識別が与えられることなどにより、この方法が悪用される可能性が小さくなる。）鍵もまた、よりセキュアに格納されることができる。任意であるが、ソフトウェアを追加して、システムがデータの使用を計測することを保証し、これを不正防止装置内に格納するようにすることも可能である。スマートカードが使用された場合、プロファイルの検査はユーザＩＤに対して行なわれ、単体の使用は、そのカードがリーダ内に残される必要がないことを意味し、場所の独立性も得られる。

【０２１７】

上述の方法Ｃを用いる使用許諾の２つの主要なオプションを参照して、まず第一の場合Ｃ１について説明する。

【０２１８】

・セキュア実行プログラムは、汎用的なものであり、ロック解除鍵の盗用を防止するため、プラットフォームに統合されている。異なるデータに同一の処理が用いられるため、これが可能であり、データ名及び関連する鍵のみが、それぞれの場合について異なることになる。セキュア実行プログラムとセキュア・ローダは、製造業社の秘密鍵を用いて署名されたハッシュされたバージョンとともに格納される。製造業社の公開鍵証明書は、いずれのプラットフォームにも含まれる。プラットフォームのブート／インストール時において、このパッケージは、ハッシュ化、及び、公開鍵証明書を用いて復号された署名との比較により、完全性が検査される。このコードは、完全性検査が失敗した場合にはロードされず、この場合、全体的なプラットフォームの完全性は欠いてる。

【０２１９】

・トラステッド・モジュールＩＤと支払いの登録時において、クリアリング・ハウスまたは開発者は、トラステッド・モジュールＩＤに対応するデータベース・エントリへ挿入されるべきデータＫのロック解除鍵を生成する（この生成は、実際には、クリアリング・ハウスまたは開発者により認証を得た第三者によって行なわれる）。

【０２２０】

・Ｃの公開鍵証明書は、Ｃによりクライアント・トラステッド・モジュールへインストールされる。Ｃからトラステッド・モジュールへの認証を含む適切なプロトコルとしては、トラステッド・モジュールにより生成されたノンスを含むトラステッド・モジュールからの認証の要求に応答して、Ｃは、Ｃの秘密鍵を用いて署名した公開鍵証明書及びノンスを含むメッセージを返す。その後、トラステッド・モジュールは、そのメッセージがＣから送られたものであることを検査することができる。

【０２２１】

保護されるべきソフトウェアまたは他のデータは、Ｋに対応する対称鍵を用いて暗号化され、Ｃの秘密コード署名鍵（たとえばマイクロソフト社のＡｕｔｈｅｎｔｉｃｏｄｅを使用して）の下で署名され、ＣによりＡのマシンのエンドユーザへ送信される。Ｋは、必要であれば、顧客ごとに異なっていてもよい。保護される必要があるのはロック解除鍵であるため、このデータは、任意の手ごろな手段（たとえばインターネットまたは衛星放送）によって、エンドユーザへ転送されることができる。

【０２２２】

・エンドユーザのプラットフォームにより受信されると、セキュア・ローダによりデータの完全性検査が実施される、この検査は、Ｃの秘密コード署名鍵に対応する公開鍵を使用して署名を検査することによって行なわれる。

【０２２３】

・完全性検査が成功すると、ソフトウェアまたは他のデータがプロットフォーム上にインストールされ、トラステッド・コンポーネントがこのイベントを記録する。成功しない場合、エラー・メッセージが発せられ、データはロードされない。

【０２２４】

・ユーザがデータを実行したい場合、セキュア実行プログラムは：

・トラステッド・モジュールＩＤを検査する。例えば、再生攻撃に対抗するためのノンスと署名された通信を含む認証により行う。；

・トラステッド・モジュールＩＤのデータベース・エントリを検査し、ロック

解除鍵Ｋを取り出す。；

・適宜、データの実行を許可または禁止する。

【０２２５】

・その後、不正防止装置は、データが実行されたか否かを記録するため、そのログを更新する。ユーザがスマートカードを用いてログインした場合、この装置のユーザＩＤが、そのデータ及び時間と共に記録される。

【０２２６】

いったんロック解除鍵が取得されたら、このロック解除鍵をデータ名と共にトラステッド・モジュール内へ格納し、そのデータについてのデータベース検索処理が再び実行される必要がないようにするという変形も可能である。データを実行するためのさらなる要求により、トラステッドＩＤを認証し、ロック解除鍵を検査し、これを用いてデータを復号し、データの実行を可能にするためのチャレンジが、ソフトウェア実行プログラムから行なわれる（上述の例Ｂと同じ方法で）。

【０２２７】

次に、データの一部を実行するためのセキュア・ライセンス許可が検査される第二の場合、Ｃ２について説明する。セキュア実行プログラム（プラットフォームに組み込まれた汎用的なコードの一部）がオペレーティング・システムと通信し、データ実行処理を開始するか否かにより、または、クリアリングハウスまたは開発者からデータの一部と共に出荷される（カスタマイズされた）ソフトウェア実行プログラムがオペレーティング・システムと通信し、データ実行処理を開始するか否かに応じて、２つの可能なサブモデルが存在する。いずれにおいても、ライセンス情報をトラステッド・モジュール自体にロードするか、または、外部データベースを参照するかについての選択がある。

【０２２８】

このモデルでは、データ自体は保護されない。データのより高い機密性が必要である場合には、例Ａまたは例Ｂの変形がかわりに使用されるべきである。

【０２２９】

第一の汎用サブモデルについて説明すると、このモデルは、例Ｃ１の鍵検査の

場合において説明したものと非常に類似している。

【０２３０】

・データベースの動作させているパーティに対応する公開鍵証明書は、クリアリングハウスまたは開発者によりインストールされる。また、その逆も言える。

【０２３１】

・エンドユーザによるデータに対する登録及び／又は支払い時において、（支払いモデルに従う）クリアリングハウスまたは開発者Ｃは、トラステッド・モジュールＩＤが伝えられる。

【０２３２】

・クライアントのトラステッド・モジュールに対応する公開鍵証明書は、クリアリングハウスまたは開発者によりインストールされる（まだ存在しない場合には）、またその逆も言える。Ｃからトラステッドモジュールへの認証を含む適切なプロトコルは、トラステッド・モジュールにより生成されたノンスを含むトラステッド・モジュールからの認証の要求に応答して、ＣはＣの秘密鍵を用いて署名されたＣの公開鍵証明書及びノンスを含むメッセージを返すことである。その後、トラステッド・モジュールはＣから来たこのメッセージを検査することができる。類似のプロトコルが、トラステッド・モジュールからＣへの公開鍵証明書の転送および認証のために用いられる。

【０２３３】

Ｃは、以下の方法により、保護されるべきアプリケーションまたは他のデータをクライアントへ送信する：データは、メッセージに添付される送信者の秘密鍵により署名されたメッセージのハッシュされたバージョンを用いて署名され、受信者がメッセージの完全性を検査できるようにしている。詳しく述べると、開発者は、関連するソフトウェア実行プログラムを伴うデータであるＭをハッシュし、これを開発者の秘密鍵（Ｓｐｒｋ）で署名して署名Σ_{Sprk}（ｈ（Ｍ））を生成する。そして、開発者はＭと共にこの署名を送信する。

【０２３４】

・その後、セキュア・ローダは、開発者の公開鍵を用いて署名を検査し、メッセージ・ハッシュを取り出す。これにより、その開発者が、その署名を検査する

のに用いられた公開鍵の持ち主であることが確認される。メッセージとメッセージ・ハッシュが得られると、セキュア・ローダは、その後、トラステッド・モジュールを介して、そのメッセージのハッシュを計算し、これを、復号化されたメッセージ・ハッシュと比較する。これは、コードの完全性を検査するものである。さらに、完全性検査機構は、ノンスの使用など、何らかの標準機構により再生攻撃を防止しなければならない。完全性検査がうまくいくと、セキュア・ローダは、データをインストールする。これにより、変更されたデータ（たとえばＡＰＩコールを持たないデータ）が実行されないこと、及び、ウィルスが侵入していないことが保証される。

【０２３５】

Ｃは、購入されたデータにしたがって、更新されるべきトラステッド・モジュールＩＤに対応するデータベース・エントリにアクセス権を与える。データベースを動作させているパーティは、共有対称鍵を設定している公開鍵暗号を用いて、及び、互いのメッセージを署名することにより、クリアリングハウスまたは開発者と通信する。保護されてべきそれぞれのメッセージのコンテンツは、ランダムに作成されたＤＥＳキーを用いて暗号化され、意図する受信者の公開鍵を用いてＲＳＡ暗号化された対称鍵とともに転送される。確実性及び完全性の検査が追加される場合には、それぞれのメッセージに対して以下のプロトコルが用いられることになる。

【０２３６】

・送信者は、ＤＥＳキーを生成する（乱数発生器を用いて生成し、これらの鍵が確実に１回しか用いられないようにする）。次に、送信者は、このキーを用いてデータＤを暗号化し、及び、受信者のＲＳＡ公開鍵を用いてＤＥＳキーを暗号化する。そして、送信者は、認証および完全性を提供するため、この情報すべてのハッシュに署名し、すべてをこの署名とともに送信する。その後、受信者のみが、ＤＥＳ暗号化鍵を復号化するためのＲＳＡ秘密鍵を有し、データＤを復号化するためにそれを使用しなければならない。

【０２３７】

ユーザからデータの一部を実行することの要求があるとき、セキュア実行プロ

グラムは、ライセンス情報を保持するデータベースを参照し、そのデータを実行するための許可が現在のプラットフォームのトラステッド・モジュールＩＤに関連づけられているか否かを確認する。関連付けられていない場合には、エラーメッセージがユーザへ生成され、データの実行は許可されない。関連付けられている場合には、セキュア実行プログラムは、データの実行をＯＳへ依頼する。

【０２３８】

次に、第二のサブモデルについて説明する。アプリケーション毎に固有のソフトウェア実行プログラムを有するこのモデルの具体例は、以下のようである。：・データに対する登録及び／又は支払い時において、（正しい支払いモデルに従う）クリアリングハウスまたは開発者Ｃは、購入されたデータに従って、更新されるべきトラステッド・モジュールＩＤに対応するデータベース・エントリにアクセス権を与える。（これに先立って、これらの団体間の公開鍵証明書が交換される：Ｃからトラステッドモジュールへの認証を含む適切なプロトコルは、トラステッド・モジュールにより生成されたノンスを含むトラステッド・モジュールからの認証の要求に応答して、Ｃが、Ｃの秘密鍵を用いて署名されたＣの公開鍵証明書及びノンスを含むメッセージを返すことである。類似のプロトコルが、トラステッド・モジュールからＣへの公開鍵証明書の転送及び認証に用いられる）。データベースを動作させるパーティは、共有対称鍵を設定している公開鍵暗号を用いて、及び、互いのメッセージに署名することにより、クリアリングハウスまたは開発者と通信する。

【０２３９】

クリアリングハウスまたは開発者は、（カスタマイズされた）ソフトウェア実行プログラムに関連づけられたデータをクライアントへ送信する。ソフトウェア実行プログラムは、トラステッド・モジュールの公開鍵がソフトウェア実行プログラムに挿入されるようにカスタマイズされる（または、セキュア実行プログラムとトラステッド・モジュールの間に共有鍵が設定される）。データとソフトウェア実行プログラムの両方がハッシュされ、クリアリングハウス／開発者の秘密鍵を用いて署名され、これに対応する公開鍵がトラステッド・モジュール上に格納される。

【０２４０】

・セキュア・ローダは、データとソフトウェア実行プログラムの完全性検査を行う。：インストール時において、ハッシュ化、および（トラステッド・モジュール内の公開鍵を用いて）復号された署名との比較により、パッケージが検証されることによって行なわれる。

【０２４１】

・デジタル署名が期待するものと合致しない場合、データとソフトウェア実行プログラムはロードされない。

【０２４２】

・ユーザがデータを実行したいと望む場合、ＯＳは、そのデータに対応するソフトウェア実行プログラムへメッセージを送信する。次に、ソフトウェア実行プログラムは、乱数（ノンス）をアプリケーションの題名と共に送信することによって、セキュア実行プログラムへチャレンジ／レスポンスを発行する。さらに、クライアント・マシンにログインするためにスマートカードＩＤが使用され、ホットデスキングが使用されるライセンスモデルである場合には、スマートカードＩＤが送信される。

【０２４３】

・セキュア実行プログラムは：

・トラステッド・モジュール内に格納されたプロファイル内のトラステッド・モジュール・マシンＩＤに基づいて、そのデータの実行が、ライセンスされているか否かを検査する。または、

・トラステッド・モジュール内に格納されたプロファイル内に挿入されたスマートカードのユーザＩＤにしたがって、そのデータの実行が、ライセンスされているか否かを検査する。または、

・外部データベースの一部を参照またはダウンロードしてトラステッド・モジュール内にプロファイルを形成し、上述の方法により、そのアプリケーションがライセンスされているか否かを確認する。

【０２４４】

・有効なライセンスが存在しない場合、セキュア実行プログラムはエラー・メ

ッセージを返し、ソフトウェア実行プログラムは、そのエラー・メッセージから使用許諾に伴う問題の正確な種類を判定して、適切にＯＳへ通知する。有効なライセンスが存在する場合、セキュア実行プログラムは、トラステッド・モジュールの秘密鍵を用いて署名および暗号化された、ノンスおよびデータ参照を含むメッセージを返す。

【０２４５】

・ソフトウェア実行プログラムは、トラステッド・モジュールの公開鍵を用いて、セキュア実行プログラムの返答が正しいか否かを検証し、データを実行するための呼び出しをＯＳへ渡すか、または、エラーメッセージをＯＳへ送信するか、のいずれかを適切に行う。

【０２４６】

例Ｄ：

第四の例は、トラステッド・モジュールの指紋によって、ドングルとしてトラステッド・モジュールを使用することである。

【０２４７】

この方法は、ハードウェア内の信用できる識別（即ち、秘密でないトラステッド・モジュール識別）、実行させるアプリケーションの完全性検査、関連するアプリケーション有効化ソフトウェアの完全性検査を使用し、ハードウェア内でセキュアな監査を使用する点において、現在の指紋照合技術とは異なる。任意であるが、ロック解除鍵は、リモートではなく、クライアント・マシン上のソフトウェア実行プログラム内で生成されることができる。トラステッド・モジュールは、鍵、保護されたデータ、及び、関連するソフトウェア実行プログラムを取得するためにベンダに接触しなければならず、これにより、トラステッド・モジュールＩＤを用いて復号化鍵をローカルに生成することが可能となる。単一の鍵がデータの復号に用いられること、または、異なる鍵がエンドユーザ毎に用いられる（こちらのほうがより安全である）ことが可能であるため、このデータは一般的に暗号化されて出荷されることができる。

【０２４８】

この方法は、Ｂの変形であり、Ｂで用いられたアプローチの代替を提供する。

これは、以下の点で異なる：

・ロック解除鍵は、リモートでなくクライアント・マシン上のソフトウェア実行プログラムまたはセキュア実行プログラム内で生成されることが可能である。；

・クリアリングハウスからクライアント・マシンへ転送される鍵はロック解除鍵ではないが、ソフトウェア実行プログラム内のアルゴリズムを用いて導出され、トラステッド・モジュールの詳細を指紋照合できる鍵である。ロック解除鍵を導出するために使用される技術は開発者間で異なる可能性があるため、セキュア実行プログラムよりもソフトウェア実行プログラムを用いる方がよい。

【０２４９】

ドングルの欠点なく、ライセンス管理システムの柔軟性を、高度のハードウェア・セキュリティに組み合わせることが可能である。この方法は、現在のライセンス保護の方法に関する問題に抗するものであり、この問題は次のようなものを含んでいる。

【０２５０】

・他のマシンを装うマシンを用いた攻撃。内部コンポーネントについての装置ＩＤであるマシンＩＤは信頼できる。これは、よりセキュアなログ記録を行う使用許諾について有用であり、より多くのライセンス情報とライセンスモデル及び認証を可能にする。装置ＩＤは、ＰＣ指紋技術に現在使用されているもの、すなわちハードディスクＩＤ，ＢＩＯＳシリアル番号，ネットワークＩＤカードなどよりも信頼性が高いため、ＰＣ指紋は現在使用されているものより偽造されにくい。そのような信頼性のある識別は、他のマシンを装うマシンを使用しての攻撃に対して役立つ。

【０２５１】

・データはバイパス（回避）または変更される可能性があり、そのためソフトウェアのみの保護では、あらゆる破壊の影響を受ける。セキュリティ、指紋照合及び認証を実施するためにとられるアクションは、ハッカーから隠される必要がある。しかしながら、すべての情報がＰＣ上に格納され、機能はＰＣのプロセッサを用いて実行されるため、これらのアクションはデバッガによりトレースされることが可能である。これらのアクションをデバッガから守る唯一の方法は、ウ

ィンドウズのＲｉｎｇ　Ｚｅｒｏのような、オペレーティング・システムまたはマシンに特化した例外を使用することである。これは大部分のデバッガをブロックすることによりセキュリティを改善するが、インテルのＰｅｎｔｉｕｍ(R)のようなＰＣプロセッサに対して広く使用可能となっているチップ・シミュレータを停止させるものではない。さらに、これは、ソフトウェアのみの解決法をマシン固有のものとし、さまざまなプラットフォーム毎にバージョンを必要とする。多くのソフトウェアのみの保護を提供するサプライヤは小規模であり、アプリケーション及び動作環境のさまざまな組み合わせのすべてに対して適時の保護モジュールを提供できない。これは、ユーザをいらつかせ、開発者のサポート時間を浪費させる非互換性につながる。いずれのプログラムがロードされる前にも、同一の認証アクションが、わずかな数の識別するＰＣコンポーネント上で実行されなければならないため、比較的わずかなコードでトレースすることができる。したがって、いったんローディング・シーケンスが解読されると、ソフトウェアのみの手段を用いるすべてのアプリケーションに対する保護は簡単に破られる可能性がある。プラットフォームとソフトウェア上での完全性検査は、関連するライセンス検査およびアップロード・ソフトウェアに対する完全性検査を可能とし、データがバイパスされたり、変更されたりすることを回避する。上述した使用許諾の態様は、ＰＣプロセッサ上に依存するものではない。－このアルゴリズム機能はトラステッド・ハードウェア内で実施されるため、デバッガまたはチップ・シミュレータによりそのプロセスが解析されることはありえない。

【０２５２】

・単一のＬＭＦは、１つの開発者により販売されたアプリケーションのすべての機能を管理できる。しかし、開発者毎に個別の構成が必要であり、おそらく異なるライセンス・マネジャー間ではクラッシュする可能性がある。ユーザサイト毎にただ１つのライセンス・マネジャーを持たせ、各開発者がこれに接続する方がよい。このモデルはより一般的で、すべての開発者をカバーできる。

【０２５３】

・ソフトウェア・ソリューションは、遅い暗号化を提供し、あまりセキュアではなく、限られた量のセキュリティのみを格納されたデータへ提供する。遅い暗

号化は、使用を限定し、すべての通信に対する大量の暗号化の使用を非実用的なものとする。エンドユーザは、自分達の通信およびアプリケーションをさらに長く待つこと、または、通信の小片のみを暗号化することのいずれかを選択できる。ハードウェア暗号化はより速い。すべての通信に対して高速の暗号化を使用することによって、通信が透過的になる。一部分暗号化より良い解決策である。ハードウェアは、不正耐性パッケージ内に収容されることができるため、よりセキュアな状態で広く認識され、そのインタフェースはよりセキュアに制御されることが可能である。ハードウェア・ソリューションにより、鍵やユーザ情報などの重要データの保護を極めて大きくすることが可能になる。

【０２５４】

例Ｄの使用の主要な種類が２つある：

・第一に、マシン・ベースのライセンスモデルが最も適切である状況において：

・データＳは鍵Ｋを用いて暗号化される。；

・ユーザはクリアリングハウスまたは開発者Ｃに登録し、相互認証があり、トラステッド・モジュールＩＤがＣに付与される。；

・Ｃは、任意の手ごろな手段により、署名されハッシュされた、暗号化されたデータおよび関連するソフトウェア実行プログラムをユーザへ送信する。；

・クライアント・コンピュータ上のセキュア・ローダが完全性を検査し、完全性検査が成功した場合、データＳをインストールする。；

・Ｃからトラステッド・モジュールへのロック解除鍵を転送には対称鍵暗号が用いられる。転送された鍵がシステムレベルのロック解除鍵である場合、この鍵は他のマシンへは有効でないため、例Ｂと同様に、第三者から保護される必要はない。；

・ソフトウェア実行プログラムは、ＣまたはＣにより信用された第三者によってその内部に事前に格納されたアルゴリズムを用いて、ロック解除鍵及びトラステッド・モジュールＩＤからＫに対応する復号化鍵を計算する。；

・復号化鍵は、データを復号化し、これを実行可能にするために用いられる。

【０２５５】

・第二に、ユーザ・ベースのライセンスモデルが必要とされる場合において：

・データＳは鍵Ｋを用いて暗号化される。；

・ユーザはクリアリングハウスまたは開発者Ｃに登録すし、相互認証がある。そして、ＣはスマートカードＩＤを与えられる。；

・Ｃは、任意の手ごろな手段により、署名されハッシュされた、暗号化されたデータ及び関連するソフトウェアをユーザへ送信する。；

・ユーザにより選択されたクライアント・コンピュータ上のセキュア・ローダは、完全性を検査し、完全性検査が成功した場合にデータＳをインストールする。；

・ロック解除鍵は、任意の手ごろな手段により、ＣからユーザへＣ転送される。この鍵は特に機密性のものではなく、電話または電子的に転送されることができる。；

・ユーザは、トラステッド・プラットフォーム・コンピュータにログインし、スマートカードをリーダへ挿入する。；

・ユーザがデータを実行しようとするとロック解除鍵の入力（タイプ）が促される。；

・ソフトウェア実行プログラムは、ＣまたはＣにより信頼された第三者によってその内部に事前に格納されたアルゴリズムを用いて、ロック解除鍵とスマートカードＩＤからＫに対応する復号化鍵を計算する。；

・この復号化鍵は、データを復号化し、データを実行可能にするために用いられる。

【０２５６】

例Ｅ：

上述の例Ａ～Ｄのいずれかを使用し、トラステッド・モジュール内で適切にセグメント化されたアプリケーションを動作させるオプションが存在する：現在と同様の方法でプラットフォーム上でアプリケーションを動作させるのことに加えて、内部マシン・トラステッド・モジュール内、または、スマートカードなどの可搬式トラステッド・モジュール内、あるいは、これらの組み合わせ内でアプリケーションを動作させるための追加のオプションがある。スマートカード上で複数のアプリケーションを動作させることについてすでに特許された当業者に既知

である最先端技術が用いられる。

【０２５７】

例Ｆ：

最後の例は、複数のトラステッド装置をどのように組み合わせたら柔軟な方法でデータの使用許諾が行えるかの例である。内部マシントラステッド・モジュールと、スマートカードなどの可搬式トラステッド・モジュールとの組み合わせは、ホットデスキング・ライセンスモデルが使用され、また、ＯＳがソフトウェア実行プログラムと通信するという特定の場合について考慮される。同様の処理は、図１９に記載のモデルに対しても使用される。

【０２５８】

データに対する登録及び／又は支払いがあると、クリアリングハウスまたは開発者（正確な支払いモデルに従う）は、購入されたデータにしたがって、更新されるトラステッド・モジュールＩＤに対応するデータベース・エントリにアクセス権を与える。（これに先立って、前の例で説明したように、相互認証があり、これらの団体間の公開鍵証明書が交換される）。データベースを動作させるパーティは、共有対称鍵を設定する公開鍵暗号化を使用して、及び、それぞれが自分のメッセージに署名することによって、クリアリングハウスまたは開発者と通信する。保護されるべきメッセージの内容は、標準のプロトコルに従って、ランダムに生成されたＤＥＳキーを用いて暗号化され、意図する受信者の公開鍵を用いてＲＳＡ暗号化された対称鍵と共にに転送される。

【０２５９】

・クリアリングハウスまたは開発者は、（カスタマイズされた）ソフトウェア実行プログラムと関連づけられたデータを、クライアントへ送信する。このソフトウェア実行プログラムは、トラステッド・モジュールの公開鍵がソフトウェア実行プログラムに挿入されるようにカスタマイズされる（または、共有鍵がセキュア実行プログラムとトラステッド・モジュールとの間に設定される）。データおよびソフトウェア実行プログラムの両方がハッシュされ、クリアリングハウス／開発者の秘密鍵を用いて署名され、これに対応する公開鍵がトラステッド・モジュールに格納される。

【０２６０】

・セキュア・ローダは、このデータおよびソフトウェア実行プログラムを完全性検査する。：これは、インストール時に、ハッシュ化、及び、（トラステッド・モジュール内の公開鍵を用いて）復号化された署名との比較により、パッケージが検証されることで行なわれる。

【０２６１】

デジタル署名が期待されるものに合致しない場合、ソフトウェア実行プログラムはロードされない。

【０２６２】

・スマートカードを使用するログオン時には、スマートカードおよびトラステッド・モジュールの公開鍵証明書が将来の通信に備えて（これがすでにされなかった場合）交換される。そして、トラステッド・モジュールとスマートカードの間に相互認証がある。

【０２６３】

・トラステッド・モジュールは（現在の）スマートカードＩＤを格納する。

【０２６４】

・ユーザが何らかのデータを実行したいと望む場合、そのデータに対応するソフトウェア実行プログラムは、そのデータへの参照と共に乱数（ノンス）を送信することにより、セキュア実行プログラムへチャレンジ／レスポンスを発行する。

【０２６５】

・セキュア実行プログラムは、スマートカードＩＤを使用して、またはスマートカード上に格納された何らかの情報を得ることによって、データに対して適切なライセンス検査を行う。たとえば、上述のライセンスモデルを使用して、このセキュア実行プログラムは：

・トラステッド・モジュール内に格納されたプロファイルに挿入されたスマートカードのユーザＩＤに従って、そのデータの実行がライセンスされているか否かを検査する。

・または、トラステッド・モジュール内に格納されたプロファイルのトラステッ

ド・モジュールＩＤに基づいて、そのデータの実行がライセンスされているか否かを検査する。または、

・外部データベースの一部を参照またはダウンロードしてトラステッド・モジュール内にプロファイルを形成し、上述の方法により、データがライセンスされているか否かを確認する。

【０２６６】

・有効なライセンスが存在しない場合、セキュア実行プログラムはエラーメッセージを返し、ソフトウェア実行プログラムは、エラーメッセージからライセンスに伴う問題の正確な種類を判定して、適切にＯＳへ知らせることができる。有効なライセンスが存在する場合、セキュア実行プログラムは、トラステッド・モジュールの秘密鍵を用いて署名および暗号化された、ノンスおよびデータ参照を含むメッセージを返す。

【０２６７】

・ソフトウェア実行プログラムは、トラステッド・モジュールの公開鍵を使用してセキュア実行プログラムの返答が正しいかを検証し、データを実行するための呼び出しをＯＳへ送るか、または、ＯＳにエラー・メッセージを送信するか、のいずれかを適切におこなう。

【０２６８】

・ログは、スマートカードではなく、マシンのトラステッド・モジュール内に保持され、適宜更新される。

【０２６９】

本発明の実施形態は純粋に例として説明しており、本発明の範囲内において多くの修正および拡張がなされることが可能であることは理解されるべきである。

【図面の簡単な説明】

【図１】

本発明の実施形態を実施可能なシステムを示す図である。

【図２】

スマートカードリーダを介してスマートカードと通信するように構成されたトラステッド装置、及び一群のモジュールを備えるマザーボードを示す図である。

【図３】

トラステッド装置をさらに詳細に示す図である。

【図４】

コンピューティング装置の完全性メトリックを取得する際に関係するステップを示す系統線図である。

【図５】

トラステッド・コンピューティング・プラットフォームと、その完全性を検証するトラステッド・プラットフォームを含むリモート・プラットフォームとの間の通信を確立する際に関係するステップを示す系統線図である。

【図６】

本発明の実施形態に従って使用されるユーザのスマートカードの動作部分を示す図である。

【図７】

スマートカードとホスト・プラットフォームとを相互認証する処理を示す系統線図である。

【図８】

図２１のシステムのトラステッド・モジュールの略ブロック図である。

【図９～図１２】

本発明で使用されるさまざまな通信方法を示す図２１のシステムの一部を示す。

【図１３】

図２１のシステムで使用されるプロトコル・データ・ユニットのフォーマットを示す。

【図１４】

本発明の特定の実施形態を説明するために用いられる図２.1のシステムの修正を示す。

【図１５】

図１４のシステムのトラステッド・モジュールの論理コンポーネントの図である。

【図１６】

図１４のシステムの保護されるデータまたはソフトウェアの構造を示す。

【図１７】

図１４のシステム上のソフトウェアまたは他のデータのインストールまたはアップグレードを示すフローチャートである。

【図１８】

ライセンス検査の１モデルを用いる図１４のシステムにおける、保護されるソフトウェアまたはデータの使用を示すフロー・チャートである。

【図１９】

ライセンス検査の他のモデルを用いる図１４のシステムにおける、保護されるソフトウェアまたはデータの使用を示すフローチャートである。

【図２０】

ライセンス検査のさらなるモデルを用いる図１４のシステムにおける、保護されるソフトウェアまたはデータの使用を示すフローチャートである。

【図２１】

他の特許出願（２０００年２月１５日付の国際特許出願第ＰＣＴ／ＣＧ００／００５０４号）の主題であるホストコンピュータ・システムの略ブロック図である。

【図１】

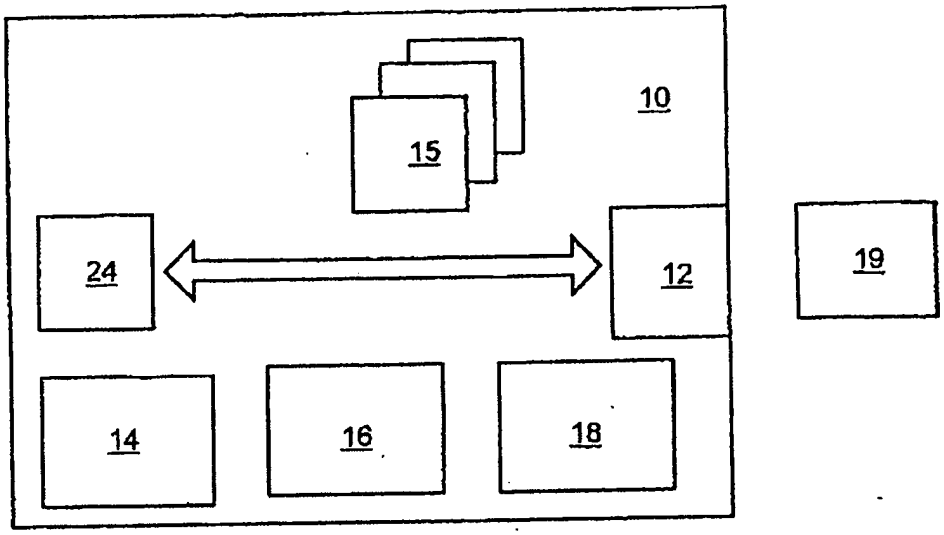


FIGURE 1

【図２】

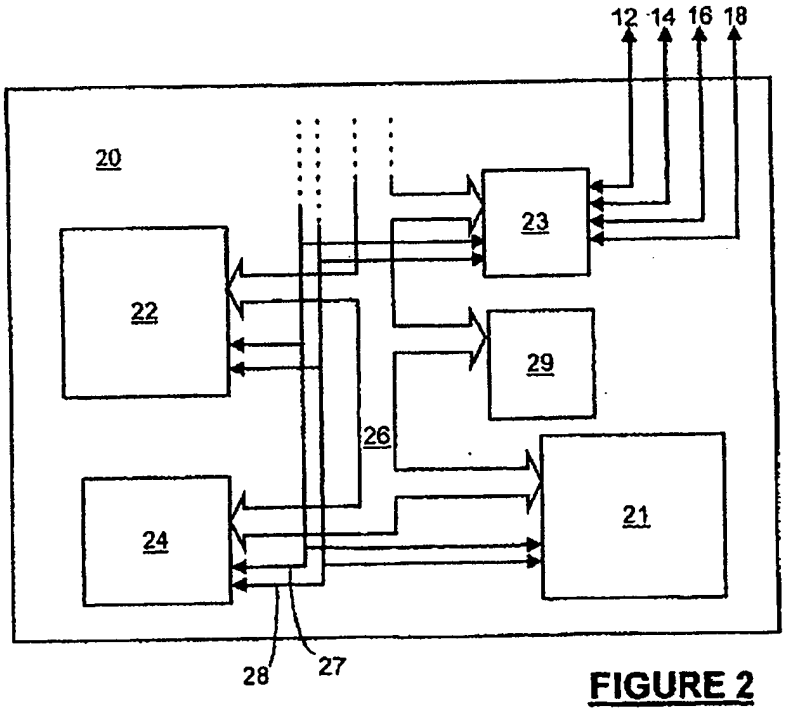


FIGURE 2

【図３】

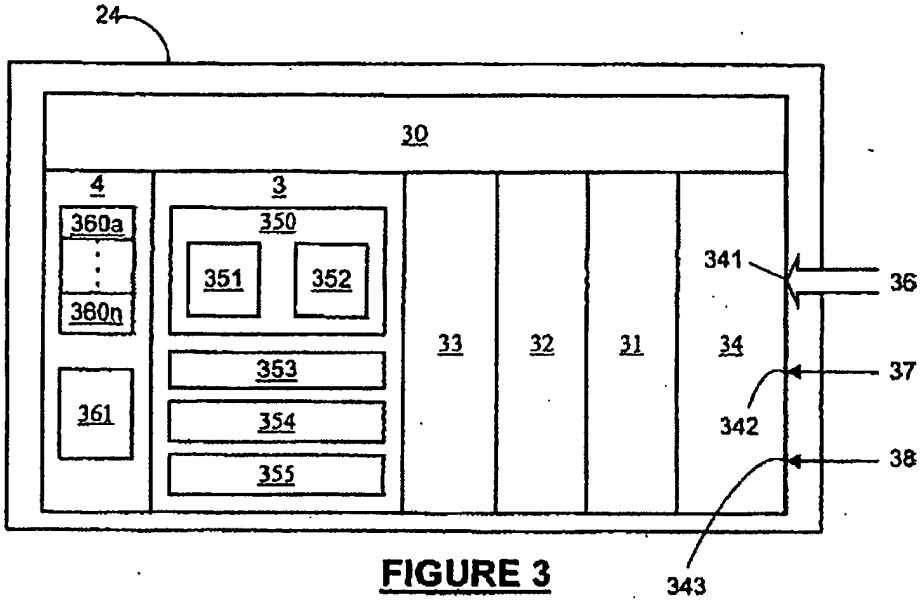
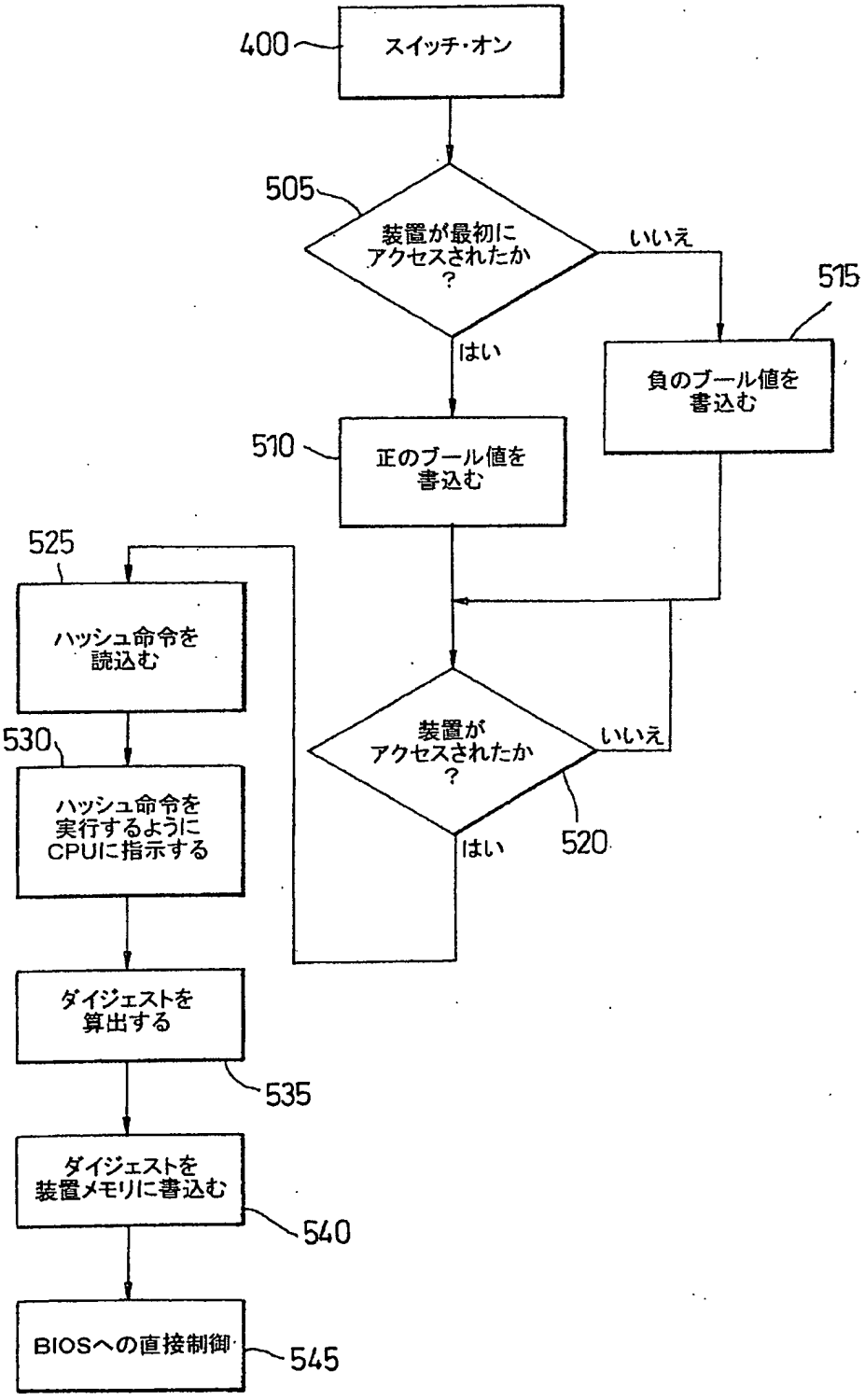
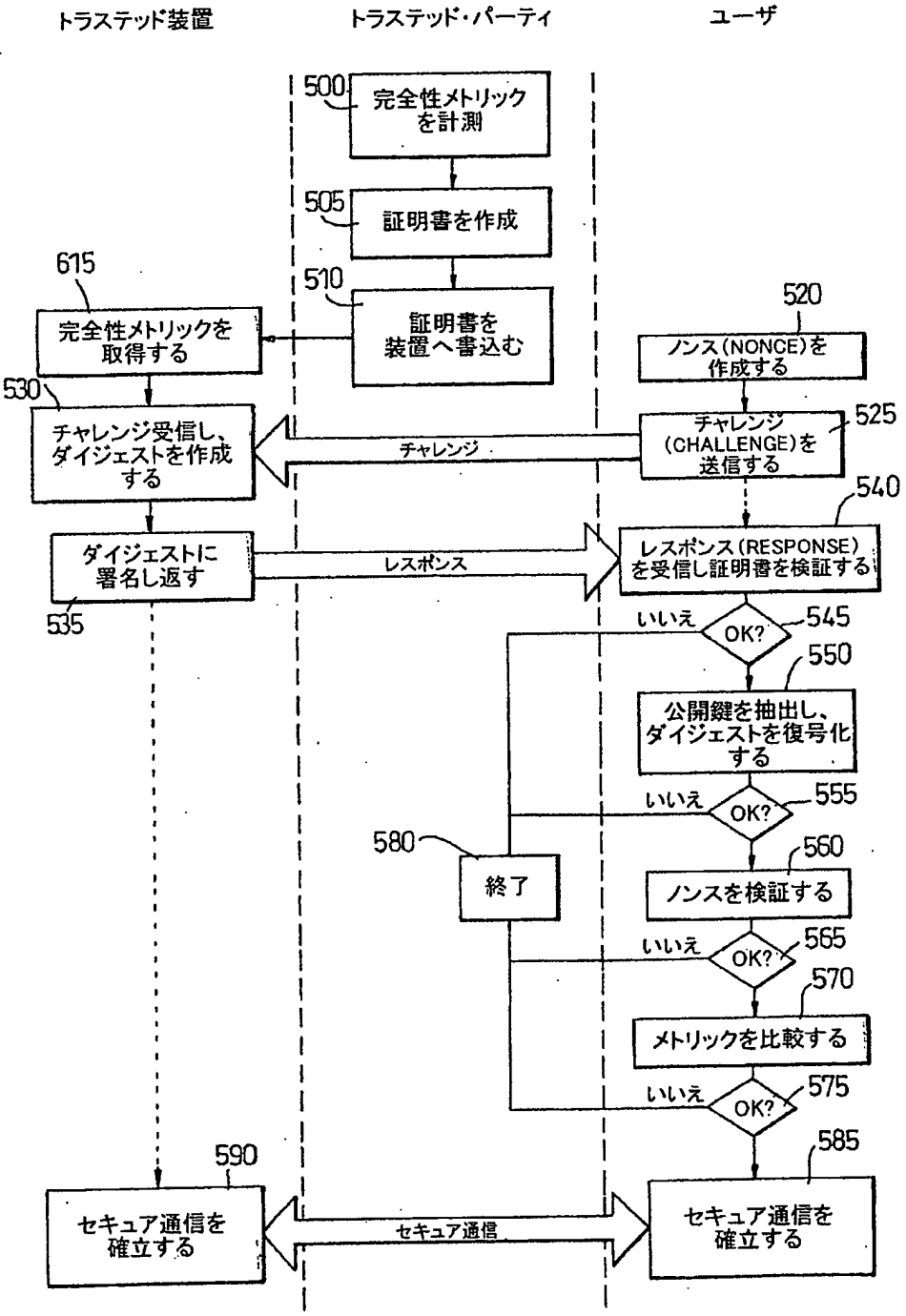

24
30
4
3
360a
360n
350
351
352
353
354
355
361
33
32
31
34
341
36
37
342
38
343

FIGURE 3

【図４】


400
スイッチ・オン
505
装置が最初にアクセスされたか？
いいえ
はい
515
負のブール値を書込む
510
正のブール値を書込む
525
ハッシュ命令を読込む
530
ハッシュ命令を実行するようにCPUに指示する
装置がアクセスされたか？
いいえ
はい
520
ダイジェストを算出する
535
ダイジェストを装置メモリに書込む
540
BIOSへの直接制御
545

【図５】


トラステッド装置
トラステッド・パーティ
ユーザ
500
完全性メトリックを計測
505
証明書を作成
510
証明書を装置へ書込む
615
完全性メトリックを取得する
520
ノンス(NONCE)を作成する
530
チャレンジ受信し、ダイジェストを作成する
チャレンジ
525
チャレンジ(CHALLENGE)を送信する
540
ダイジェストに署名し返す
レスポンス
レスポンス(RESPONSE)を受信し証明書を検証する
535
いいえ
OK?
545
550
公開鍵を抽出し、ダイジェストを復号化する
いいえ
OK?
555
580
終了
560
ノンスを検証する
いいえ
OK?
565
570
メトリックを比較する
いいえ
OK?
575
585
590
セキュア通信を確立する
セキュア通信
セキュア通信を確立する

【図６】

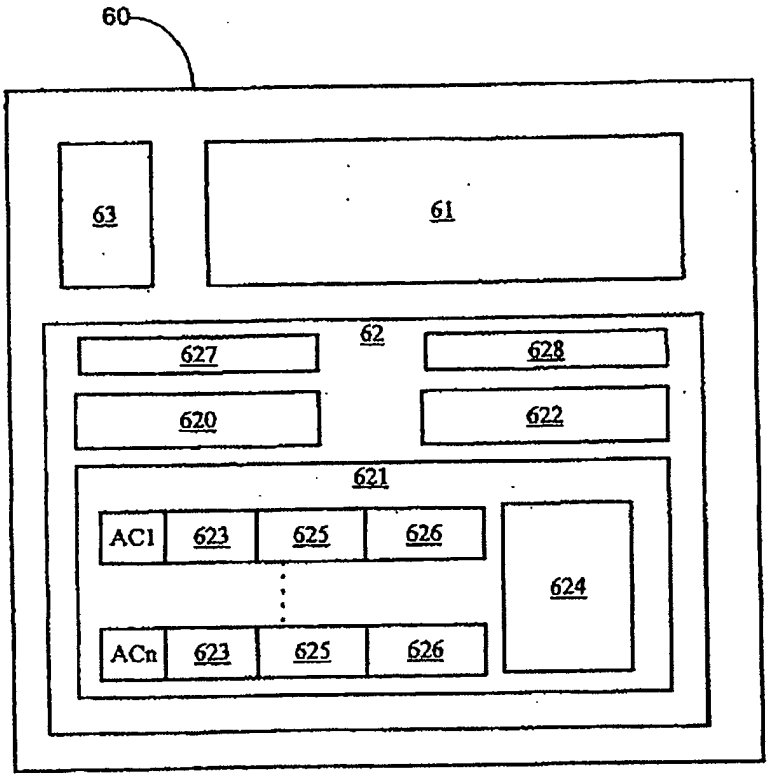
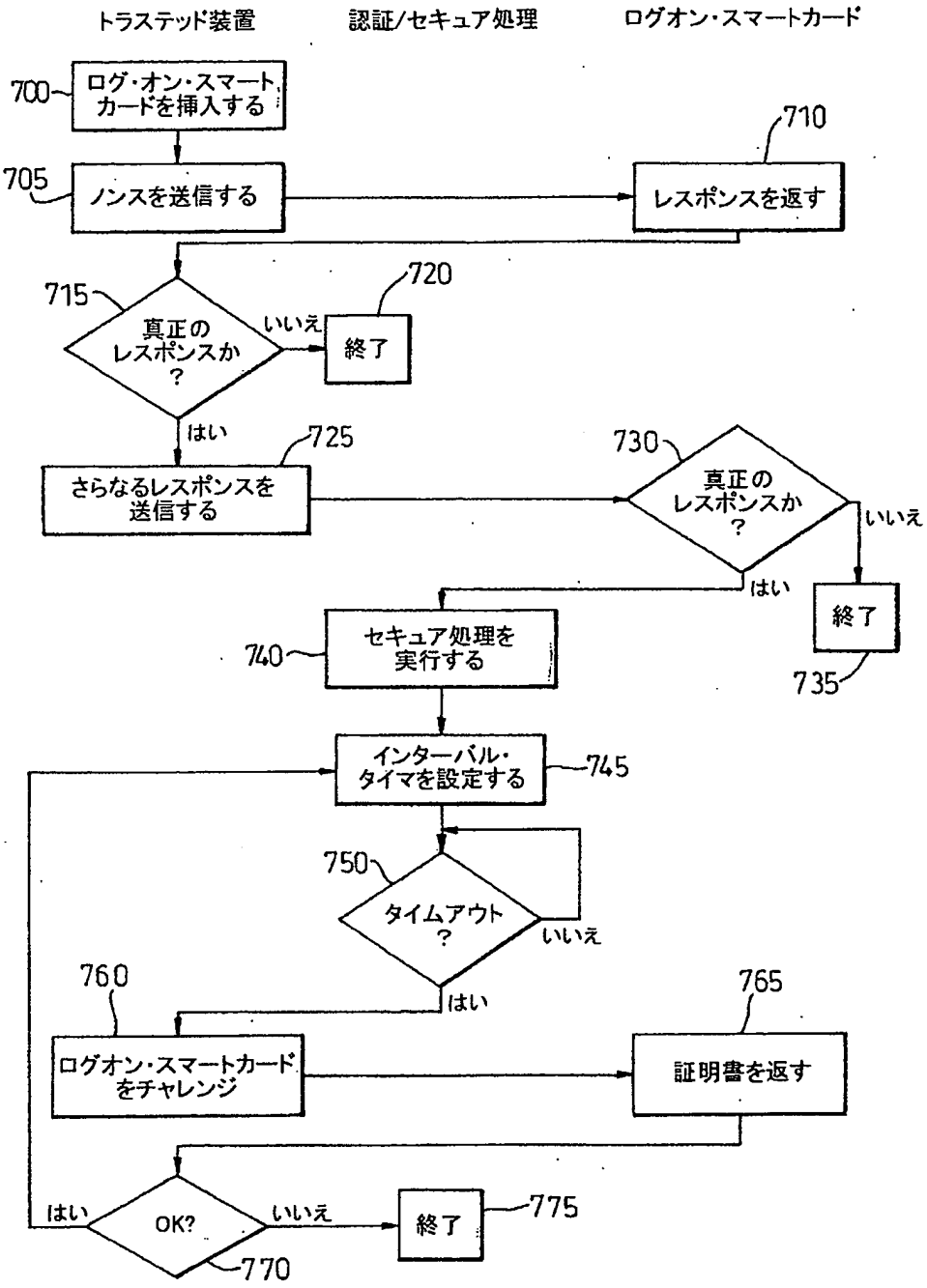
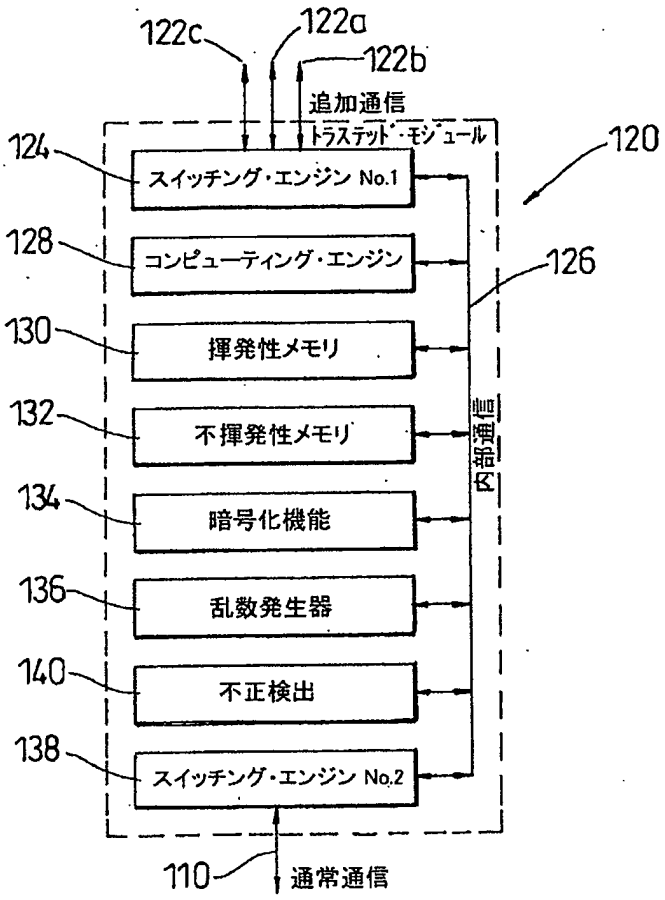

60
63
61
62
627
628
620
622
621
AC1
623
625
626
624
ACn
623
625
626

FIGURE 6

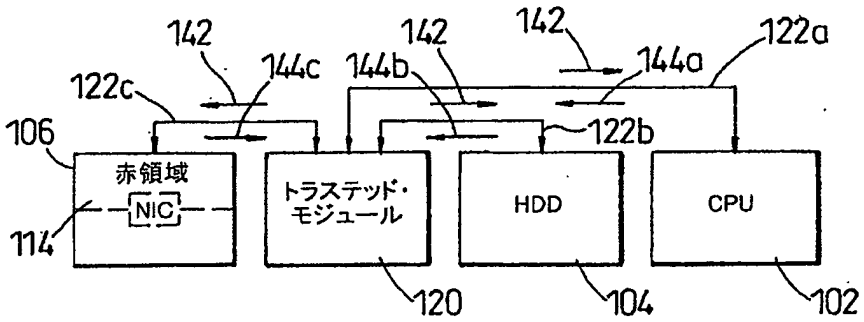
【図７】


トラステッド装置
認証/セキュア処理
ログオン・スマートカード
700
ログ・オン・スマートカードを挿入する
705
ノンスを送信する
710
レスポンスを返す
715
真正のレスポンスか？
いいえ
720
終了
はい
725
さらなるレスポンスを送信する
730
真正のレスポンスか？
いいえ
はい
終了
735
740
セキュア処理を実行する
インターバル・タイマを設定する
745
750
タイムアウト？
いいえ
はい
760
ログオン・スマートカードをチャレンジ
765
証明書を返す
はい
OK?
いいえ
終了
770
775

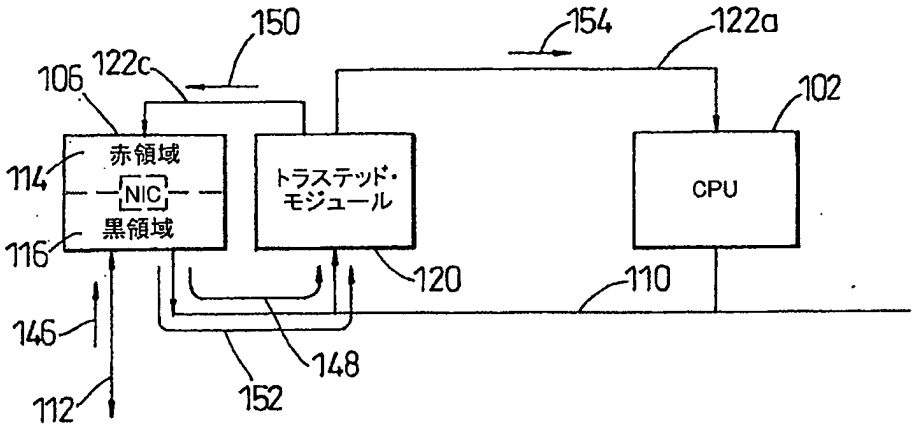
【図８】


122c
122a
122b
追加通信
トラステッド・モジュール
120
124
スイッチング・エンジン No.1
128
コンピューティング・エンジン
126
130
揮発性メモリ
132
不揮発性メモリ
内部通信
134
暗号化機能
136
乱数発生器
140
不正検出
138
スイッチング・エンジン No.2
110
通常通信

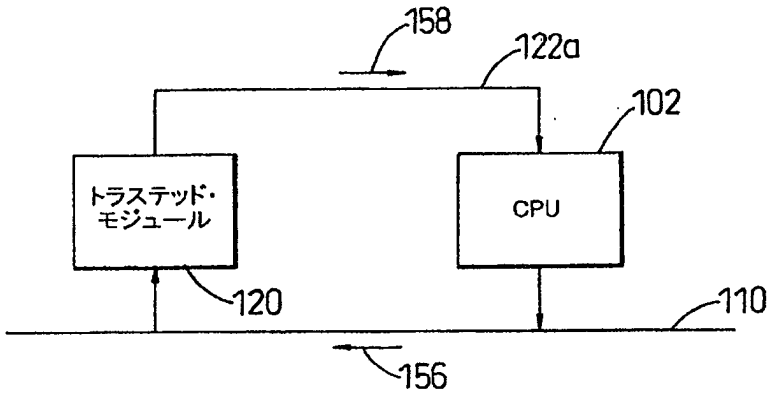
【図９】


142
142
142
122a
144c
144b
144a
122c
106
122b
赤領域
NIC
114
トラステッド・モジュール
HDD
CPU
120
104
102

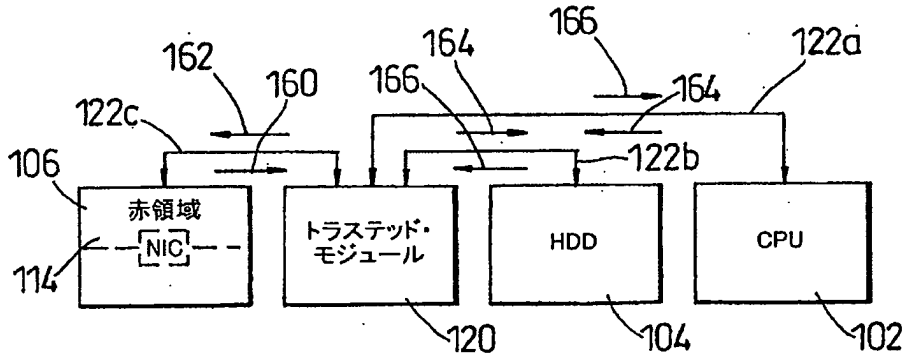
【図１０】


150
154
122a
106
122c
102
赤領域
NIC
黒領域
トラステッド・モジュール
CPU
114
116
120
110
146
148
152
112

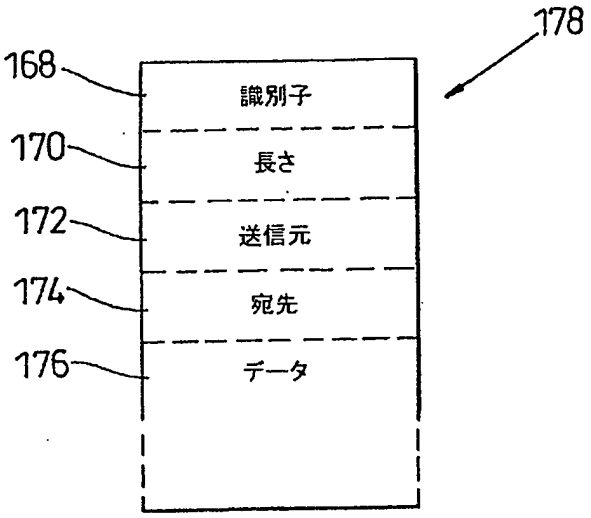
【図１１】


158
122a
102
トラステッド・モジュール
CPU
120
110
156

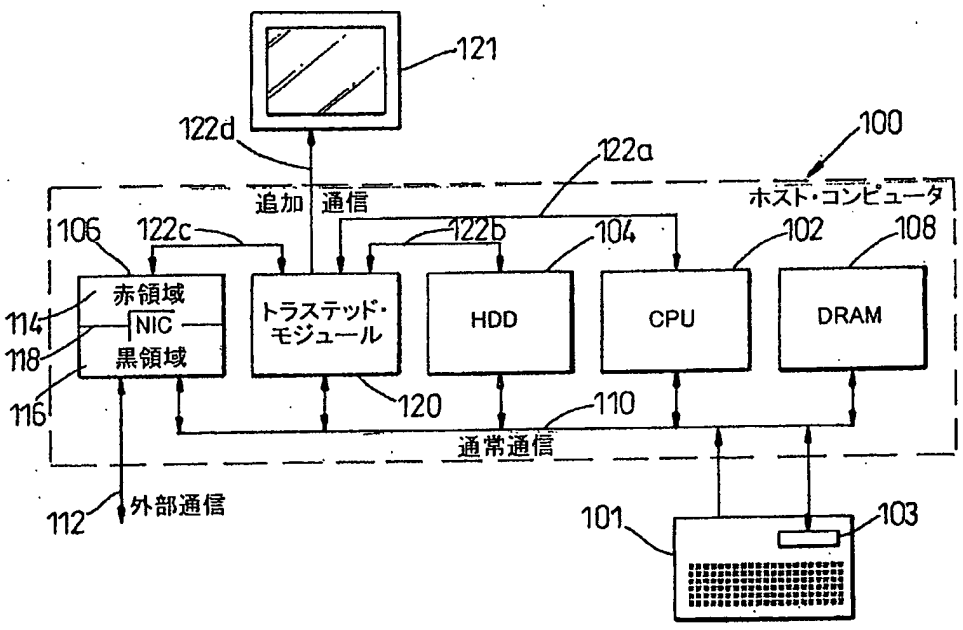
【図１２】


166
162
164
122a
160
166
164
122c
122b
106
赤領域
NIC
トラステッド・モジュール
HDD
CPU
114
120
104
102

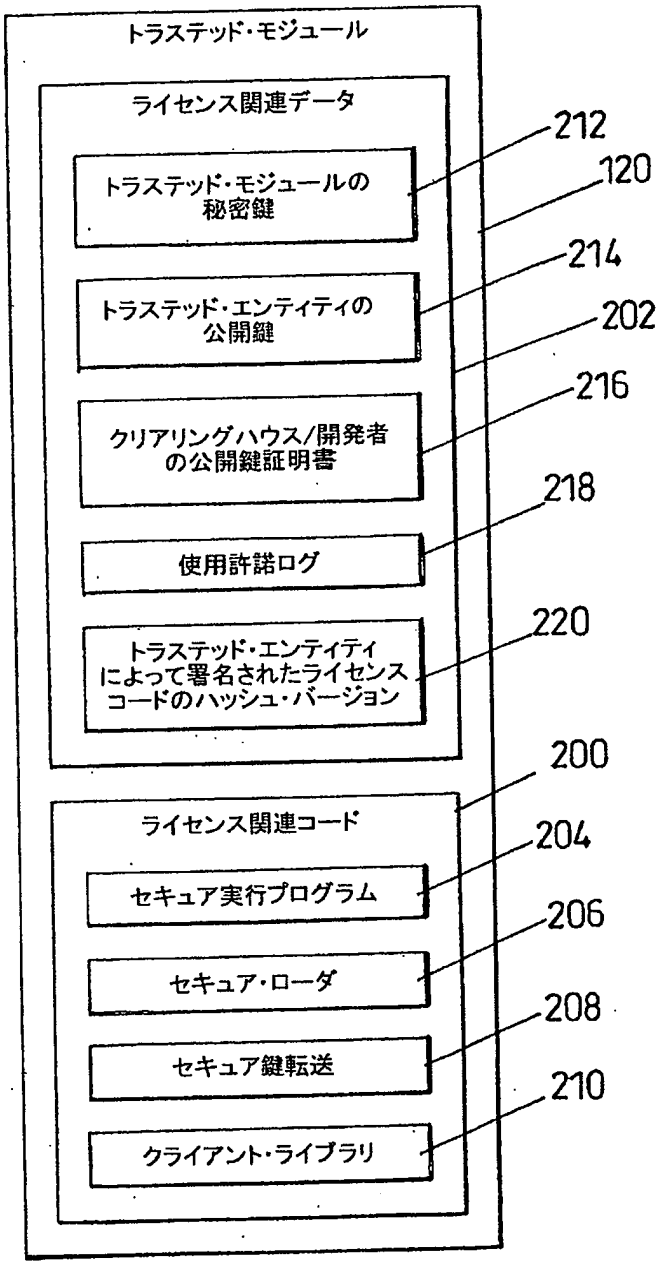
【図１３】


178
168
識別子
170
長さ
172
送信元
174
宛先
176
データ

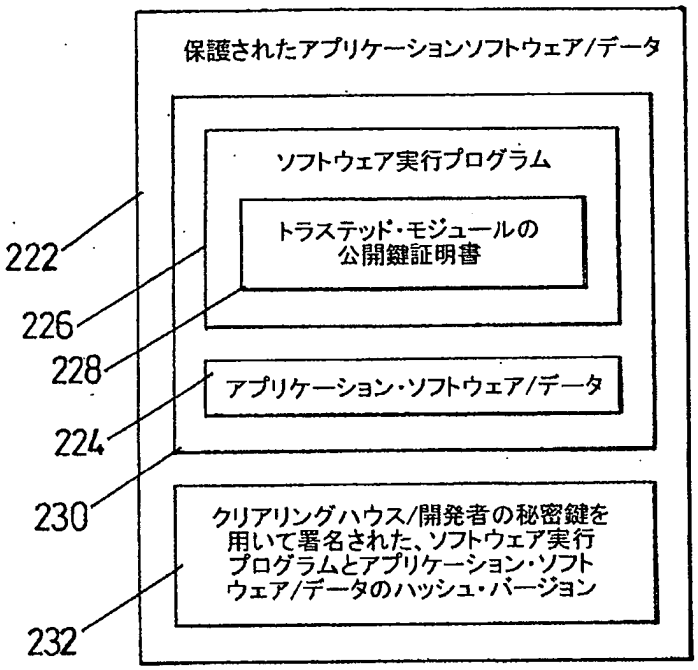
【図１４】


121
122d
100
122a
追加 通信
ホスト・コンピュータ
106
122c
122b
104
102
108
赤領域
NIC
黒領域
トラステッド・モジュール
HDD
CPU
DRAM
114
118
116
120
110
通常通信
112
外部通信
101
103

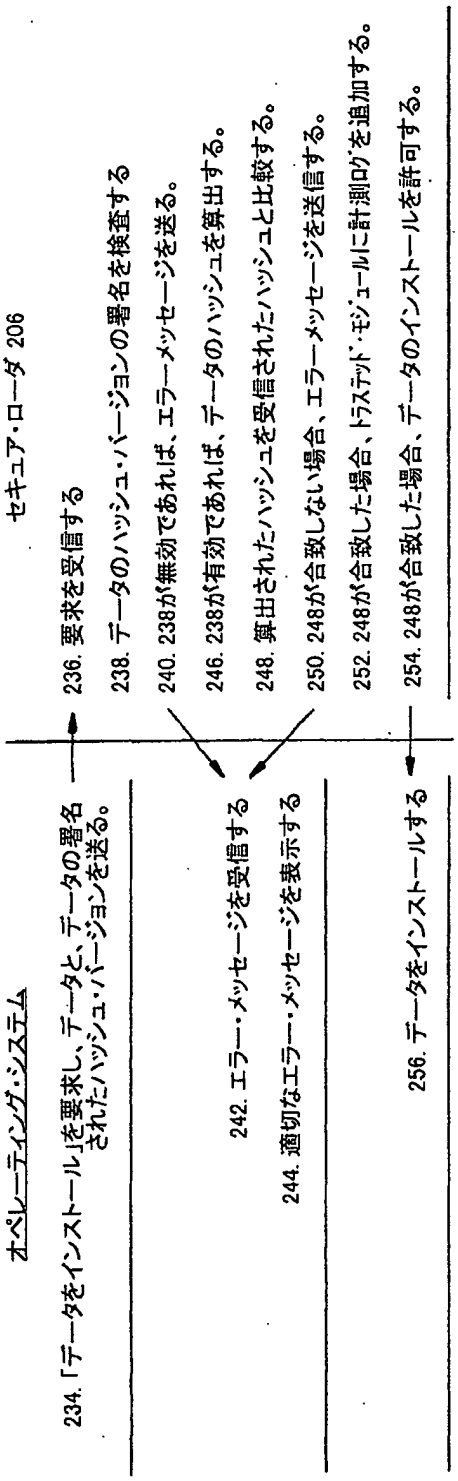
【図１５】


トラステッド・モジュール
ライセンス関連データ
トラステッド・モジュールの秘密鍵
トラステッド・エンティティの公開鍵
クリアリングハウス/開発者の公開鍵証明書
使用許諾ログ
トラステッド・エンティティによって署名されたライセンスコードのハッシュ・バージョン
ライセンス関連コード
セキュア実行プログラム
セキュア・ローダ
セキュア鍵転送
クライアント・ライブラリ
212
120
214
202
216
218
220
200
204
206
208
210

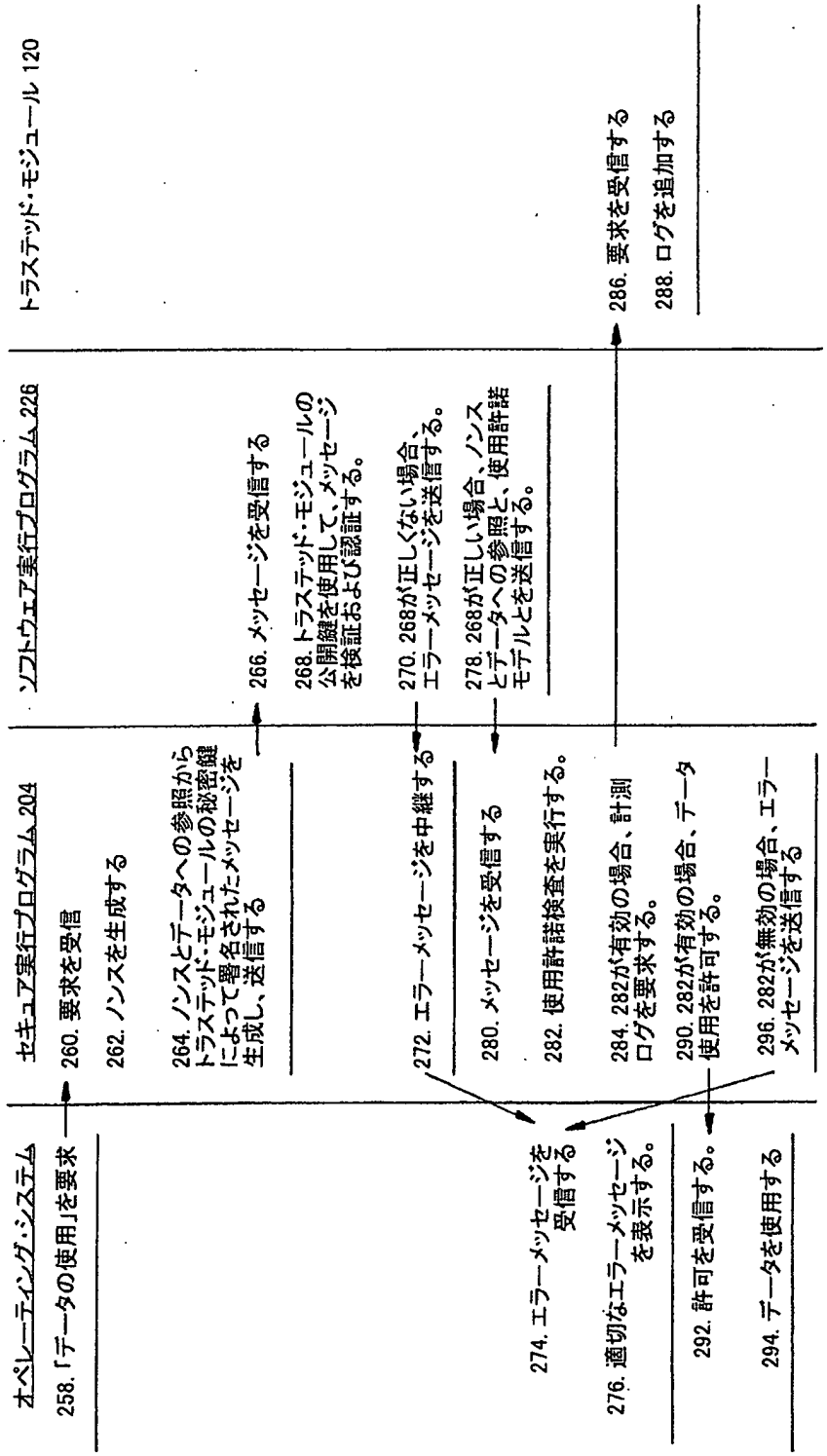
【図１６】


保護されたアプリケーションソフトウェア/データ
ソフトウェア実行プログラム
トラステッド・モジュールの
公開鍵証明書
アプリケーション・ソフトウェア/データ
クリアリングハウス/開発者の秘密鍵を
用いて署名された、ソフトウェア実行
プログラムとアプリケーション・ソフト
ウェア/データのハッシュ・バージョン
222
226
228
224
230
232

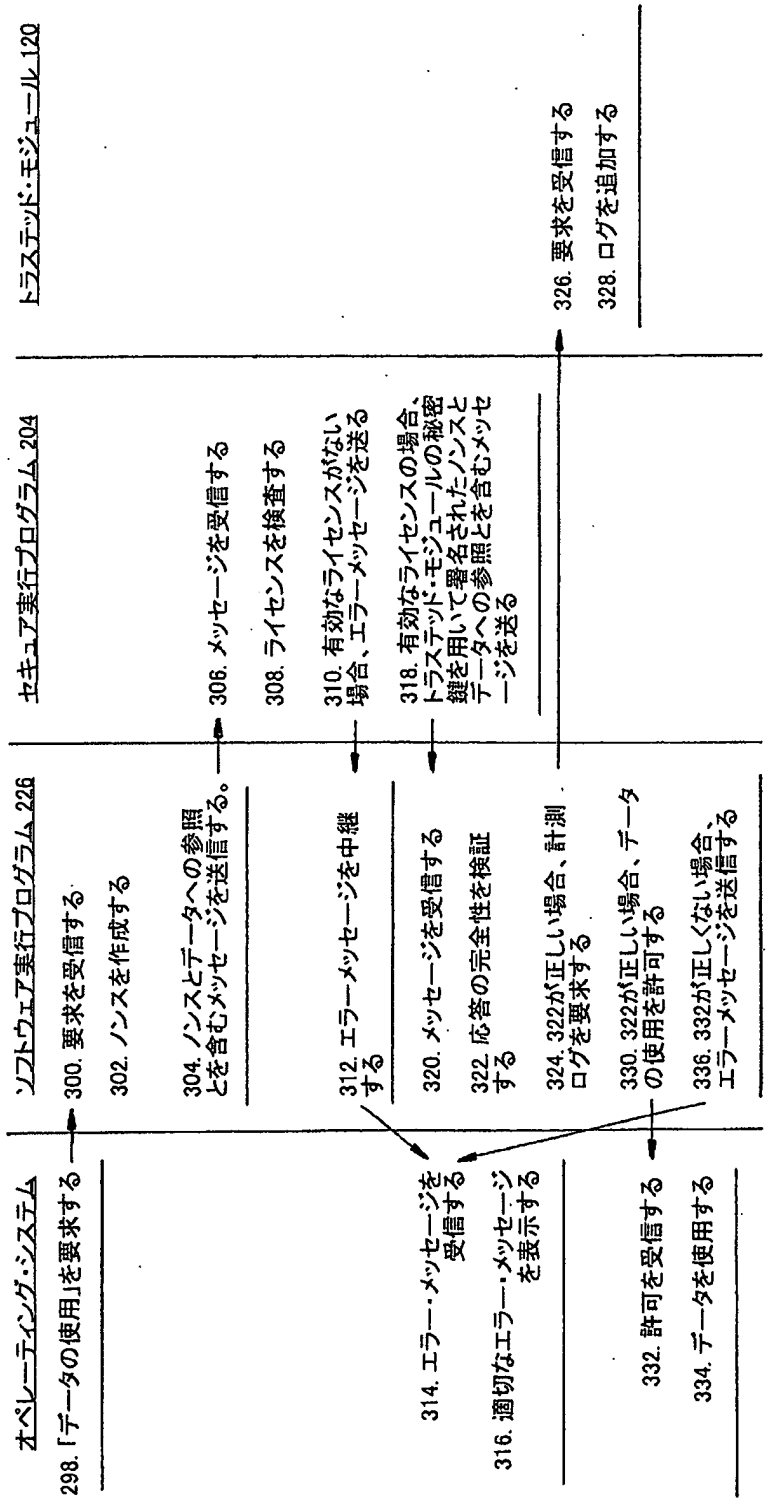
【図１７】


オペレーティング・システム
セキュア・ローダ 206
234.「データをインストール」を要求し、データと、データの署名されたハッシュ・バージョンを送る。
236. 要求を受信する
238. データのハッシュ・バージョンの署名を検査する
240. 238が無効であれば、エラー・メッセージを送る。
242. エラー・メッセージを受信する
244. 適切なエラー・メッセージを表示する
246. 238が有効であれば、データのハッシュを算出する。
248. 算出されたハッシュを受信されたハッシュと比較する。
250. 248が合致しない場合、エラー・メッセージを送信する。
252. 248が合致した場合、トラステッド・モジュールに計測ログを追加する。
254. 248が合致した場合、データのインストールを許可する。
256. データをインストールする

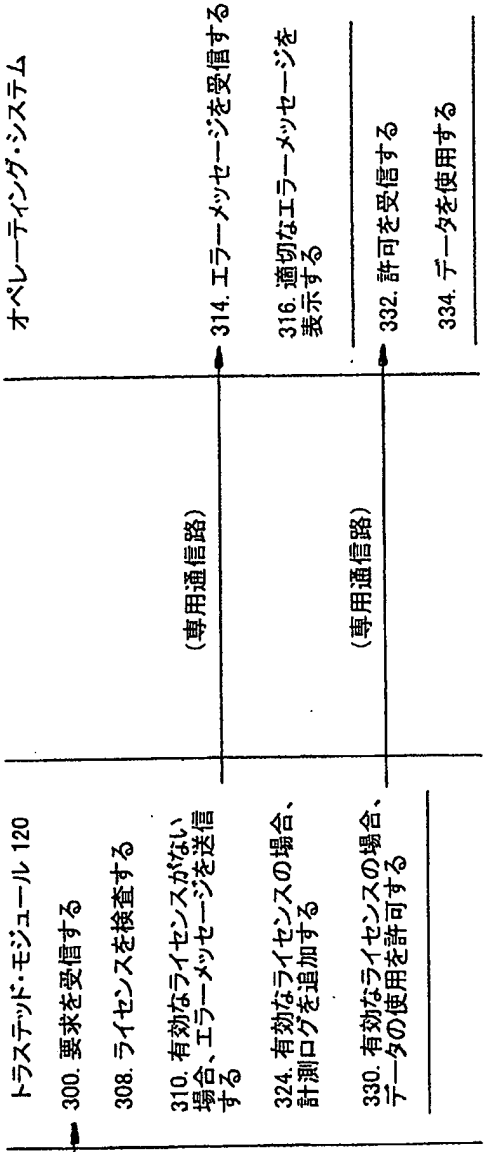
【図１８】


オペレーティング・システム
セキュア実行プログラム 204
ソフトウェア実行プログラム 226
トラステッド・モジュール 120
258.「データの使用」を要求
260. 要求を受信
262. ノンスを生成する
264. ノンスとデータへの参照からトラステッド・モジュールの秘密鍵によって署名されたメッセージを生成し、送信する
266. メッセージを受信する
268. トラステッド・モジュールの公開鍵を使用して、メッセージを検証および認証する。
270. 268が正しくない場合、エラーメッセージを送信する。
272. エラーメッセージを中継する
278. 268が正しい場合、ノンスとデータへの参照と、使用許諾モデルとを送信する。
280. メッセージを受信する
282. 使用許諾検査を実行する。
284. 282が有効の場合、計測ログを要求する。
286. 要求を受信する
288. ログを追加する
274. エラーメッセージを受信する
276. 適切なエラーメッセージを表示する。
290. 282が有効の場合、データ使用を許可する。
292. 許可を受信する。
296. 282が無効の場合、エラーメッセージを送信する
294. データを使用する

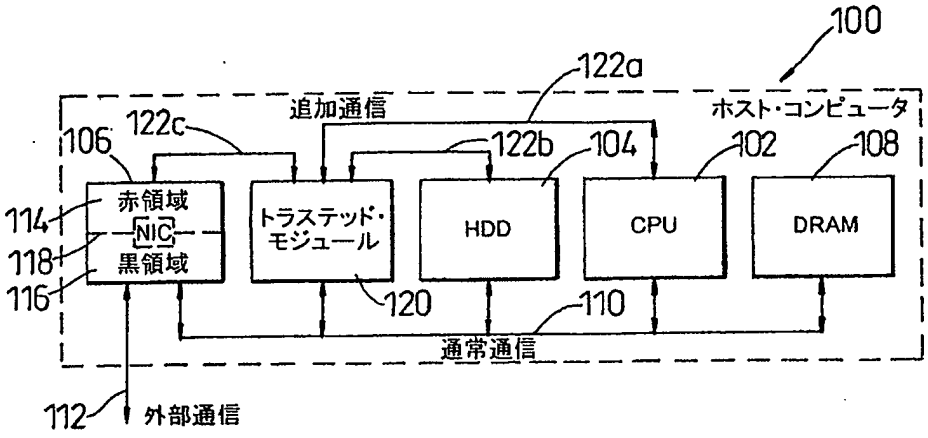
【図１９】


オペレーティング・システム
ソフトウェア実行プログラム 226
セキュア実行プログラム 204
トラステッド・モジュール 120
298.「データの使用」を要求する
300. 要求を受信する
302. ノンスを作成する
304. ノンスとデータへの参照と含むメッセージを送信する。
306. メッセージを受信する
308. ライセンスを検査する
310. 有効なライセンスがない場合、エラーメッセージを送る
312. エラーメッセージを中継する
314. エラー・メッセージを受信する
316. 適切なエラー・メッセージを表示する
318. 有効なライセンスの場合、トラステッド・モジュールの秘密鍵を用いて署名されたノンスとデータへの参照とを含むメッセージを送る
320. メッセージを受信する
322. 応答の完全性を検証する
324. 322が正しい場合、計測ログを要求する
326. 要求を受信する
328. ログを追加する
330. 322が正しい場合、データの使用を許可する
332. 許可を受信する
334. データを使用する
336. 332が正しくない場合、エラーメッセージを送信する

【図２０】


ソフトウェア実行プログラム 226
トラステッド・モジュール 120
オペレーティング・システム
298.「データの使用」を要求する（任意で、ライセンス検査を指定する）
300. 要求を受信する
308. ライセンスを検査する
310. 有効なライセンスがない場合、エラーメッセージを送信する
314. エラーメッセージを受信する
316. 適切なエラーメッセージを表示する
（専用通信路）
324. 有効なライセンスの場合、計測ログを追加する
330. 有効なライセンスの場合、データの使用を許可する
332. 許可を受信する
334. データを使用する
（専用通信路）

【図２１】


100
122a
追加通信
ホスト・コンピュータ
106
122c
122b
104
102
108
赤領域
NIC
黒領域
114
118
116
トラステッド・
モジュール
HDD
CPU
DRAM
120
110
通常通信
112
外部通信

【国際調査報告】

INTERNATIONAL SEARCH REPORT

Int ernal Application No
PCT/GB 00/03101

A. CLASSIFICATION OF SUBJECT MATTER
IPC 7 G06F1/00

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
IPC 7 G06F

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practical, search terms used)
WPI Data, EPO-Internal, PAJ

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category *	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
Y	US 5 473 692 A (DAVIS DEREK L) 5 December 1995 (1995-12-05) the whole document	1-3,6,28
A		4,5,7-27
Y	EP 0 684 538 A (IBM) 29 November 1995 (1995-11-29) column 5, line 6 - line 53	1-3,6,28
A	US 5 680 547 A (CHANG STEVE MING-JANG) 21 October 1997 (1997-10-21)	
A	WO 98 36517 A (JPC INC) 20 August 1998 (1998-08-20)	
A	EP 0 849 657 A (NCR INT INC) 24 June 1998 (1998-06-24)	

☐ Further documents are listed in the continuation of box C. ☒ Patent family members are listed in annex.

* Special categories of cited documents :

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier document but published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art.

"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
26 September 2000	04/10/2000

Name and mailing address of the ISA	Authorized officer
European Patent Office, P.B. 5818 Patentlaan 2 NL – 2280 HV Rijswijk Tel. (+31-70) 340-2040, Tx. 31 651 epo nl, Fax (+31-70) 340-3016	Powell, D

Form PCT/ISA/210 (second sheet) (July 1992)

INTERNATIONAL SEARCH REPORT

information on patent family members

Int... al Application No
PCT/GB 00/03101

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
US 5473692 A	05-12-1995	AU 3583295 A	27-03-1996
		EP 0780039 A	25-06-1997
		JP 10507324 T	14-07-1998
		WO 9608092 A	14-03-1996
		US 5568552 A	22-10-1996
EP 0684538 A	29-11-1995	US 5564038 A	08-10-1996
		JP 7319689 A	08-12-1995
		US 5771347 A	23-06-1998
US 5680547 A	21-10-1997	US 5444850 A	22-08-1995
		AU 1042895 A	15-05-1996
		JP 10511783 T	10-11-1998
		WO 9613002 A	02-05-1996
WO 9836517 A	20-08-1998	US 5953502 A	14-09-1999
		EP 1013023 A	28-06-2000
		US 6038667 A	14-03-2000
EP 0849657 A	24-06-1998	JP 10282884 A	23-10-1998
		ZA 9710559 A	24-05-1999

Form PCT/ISA/210 (patent family annex) (July 1992)

フロントページの続き

(72)発明者 チャン，デイビッド
アメリカ合衆国カリフォルニア州95030，
モンテセレノ，メイズ・アベニュー・
16112

Ｆターム(参考) 5B017 AA07 BA05 BA07 CA16
5B076 BB06 FB01 FB10

【公報種別】特許法第１７条の２の規定による補正の掲載
【部門区分】第６部門第３区分
【発行日】平成17年12月22日(2005.12.22)

【公表番号】特表2003-507785(P2003-507785A)
【公表日】平成15年2月25日(2003.2.25)
【出願番号】特願2001-517235(P2001-517235)
【国際特許分類第７版】
　Ｇ０６Ｆ　1/00
　Ｇ０６Ｆ　12/14
【ＦＩ】
　Ｇ０６Ｆ　9/06　６６０Ａ
　Ｇ０６Ｆ　12/14　３２０Ａ
　Ｇ０６Ｆ　9/06　６６０Ｈ

【手続補正書】
【提出日】平成16年11月8日(2004.11.8)
【手続補正１】
【補正対象書類名】明細書
【補正対象項目名】特許請求の範囲
【補正方法】変更
【補正の内容】
【特許請求の範囲】
【請求項１】
内部の不正に対する抵抗力を有し、第三者の公開鍵証明書(214)を格納するトラステッド・モジュール(120)と、
コンピュータプラットフォームまたは該コンピュータプラットフォームのユーザが特定のデータを使用するための許諾を受けているか否かを検査し、該データを使用するためのインタフェース、該データの使用を監視するためのインタフェース、またはそれら両方のインタフェースとして機能するセキュア実行プログラム(204)、および前記プラットフォームまたは該プラットフォームのユーザが特定のデータをインストールするための許諾を受けているか否かの検査、インストール前のデータの完全性の検査、またはそれら両方の検査を実施するためのセキュア・ローダ(206)のうちの少なくとも一方を含むライセンス関連コードを格納する手段と、
第三者の秘密鍵を用いて署名された前記ライセンス関連コードのハッシュされたバージョンを格納する手段とからなるコンピュータ・プラットフォームであって、
前記プラットフォームのブート時に、前記署名されたバージョンおよび前記公開鍵証明書(214)を参照して、前記ライセンス関連コード(200)の完全性検査を実施し、該完全性検査が失敗した場合、前記ライセンス関連コード(200)をロードしないようにプログラムされた、コンピュータ・プラットフォーム。
【請求項２】
前記完全性検査は、
前記ライセンス関連コード(200)を読み込み、該ライセンス関連コード(200)をハッシュして第１のハッシュを生成し、
前記署名されたバージョンを読み込み、該署名されたバージョンを前記公開鍵証明書(214)を用いて復号して第２のハッシュを生成し、
前記第１のハッシュと前記第２のハッシュを比較することによって実施される、請求項１に記載のコンピュータ・プラットフォーム。
【請求項３】
前記ライセンス関連コード(200)は、前記トラステッド・モジュール(120)と、他のコン

ピュータ・プラットフォームの他のトラステッド・モジュール(120)との間で、ライセンス鍵を転送できるようにするためのセキュア鍵転送コード(208)をさらに含む、請求項１または請求項２に記載のコンピュータ・プラットフォーム。

【請求項４】

前記ライセンス関連コード(200)は、前記トラステッド・モジュール(120)と通信するために呼び出されるインタフェース・サブルーチンのライブラリ(210)をさらに含む、請求項１～４のうちのいずれか一項に記載のコンピュータ・プラットフォーム。

【請求項５】

前記ライセンス関連コード(200)は、少なくとも１グループのデータについて、対応するデータグループを指定し、該データグループに対するインタフェースとして機能するソフトウェア実行プログラムを（それぞれ）含む、請求項１～４のうちのいずれか一項に記載のコンピュータ・プラットフォーム。

【請求項６】

前記ライセンス関連コード(200)を格納する手段、および前記ライセンス関連コード(200)のハッシュされたバージョンを格納する手段のうちのいずれか一方または両方の少なくとも一部が、前記トラステッド・モジュール(120)によって提供される、請求項１～５のうちのいずれか一項に記載のコンピュータ・プラットフォーム。

【請求項７】

前記トラステッド・モジュール(120)と前記プラットフォームのオペレーティング・システムとの間に、前記コンピュータ・プラットフォームの他の部分へはアクセスすることができない専用通信路(122a)を有する、請求項１～６のうちのいずれか一項に記載のコンピュータ・プラットフォーム。

【請求項８】

前記オペレーティング・システムは、前記プラットフォームまたは該プラットフォームのユーザが前記特定のデータをインストールするための許諾を受けているか否かに関するライセンス検査、および前記特定のデータの完全性の検査のうちのいずれか一方または両方を前記セキュア・ローダ(206)に対して要求するように動作し、

前記セキュア・ローダ(206)は、前記要求に応答してその検査を実施し、該検査の結果を前記オペレーティング・システムに返答するように動作し、

前記オペレーティング・システムは、前記返答に応じて前記特定のデータをインストールするか否かを決めるように動作する、請求項１～７のうちのいずれか一項に記載のコンピュータ・プラットフォーム。

【請求項９】

前記オペレーティング・システムは、前記セキュア・ローダ(206)にのみ応答して前記特定のデータをインストールするようにプログラムされる、請求項８に記載のコンピュータ・プラットフォーム。

【請求項１０】

前記トラステッド・モジュール(120)は、インストールされる前記特定のデータに関連するパーティの公開鍵証明書(216)を格納し、

前記オペレーティング・システムは、前記検査の要求の中に、前記特定のデータと一緒に、前記関連するパーティの秘密鍵を用いて署名された前記特定のデータのハッシュされたバージョンを含めるように動作し、

前記検査を実施する際に、前記セキュア・ローダ(206)は、前記要求に含まれる前記特定のデータをハッシュして第３のハッシュを生成し、前記要求に含まれる前記ハッシュされ署名されたバージョンを前記関連するパーティの公開鍵証明書(216)を用いて復号して第４のハッシュを生成し、前記第３のハッシュと前記第４のハッシュが一致するか否かに応じて前記応答を生成するように動作する、請求項８または請求項９に記載のコンピュータ・プラットフォーム。

【請求項１１】

前記検査の要求は、前記特定のデータのための前記ソフトウェア実行プログラムを含む

、請求項５に直接または間接に従属したときの請求項１０に記載のコンピュータ・プラットフォーム。

【請求項１２】

前記ソフトウェア実行プログラム（または前記ソフトウェア実行プログラムのうちの少なくとも１つ）は、特定のデータのインストールを前記トラステッド・モジュール(120)に対して要求するように動作し、

前記トラステッド・モジュール(120)内の前記セキュア・ローダ(206)は、前記要求に応答して、前記プラットフォームまたは該プラットフォームのユーザが前記特定のデータをインストールするための許諾を受けているか否かに関するライセンス検査、および前記データの完全性の検査のうちのいずれか一方または両方を実施し、該検査の結果を前記オペレーティング・システムに返答するように動作し、

前記オペレーティング・システムは、前記応答に応じて前記特定のデータをインストールするか否かを決めるように動作する、請求項５に従属したときの請求項６、および該請求項６に従属したときの請求項７～１１のうちのいずれか一項に記載のコンピュータ・プラットフォーム。

【請求項１３】

前記オペレーティング・システムは、前記トラステッド・モジュール(120)にのみ応答して前記特定のデータをインストールするようにプログラムされる、請求項１２に記載のコンピュータ・プラットフォーム。

【請求項１４】

前記トラステッド・モジュール(120)から前記オペレーティング・システムへの前記応答は、前記専用通信路(122a)を介して供給される、請求項７に従属したときの、請求項１２または請求項１３に記載のコンピュータ・プラットフォーム。

【請求項１５】

前記検査が成功した場合、前記トラステッド・モジュール(120)が、前記特定のデータを監査するためのログを生成するように動作する、請求項８～１４のうちのいずれか一項に記載のコンピュータ・プラットフォーム。

【請求項１６】

前記検査が成功した場合、前記セキュア・ローダ(206)が、前記特定のデータに対してウィルス検査を実施するように動作する、請求項８～１５のうちのいずれか一項に記載のプラットフォーム。

【請求項１７】

インストールの際に、前記特定のデータが前記トラステッド・モジュール(120)にインストールされる、請求項８～１６のうちのいずれか一項に記載のコンピュータ・プラットフォーム。

【請求項１８】

着脱可能な他のトラステッド・モジュール(120)をさらに含み、

最初に述べたトラステッド・モジュール(120)と前記着脱可能なトラステッド・モジュール(19)との間で認証検査を実施するように動作し、

インストールの際に、前記特定のデータが、前記他のトラステッド・モジュール(120)にインストールされる、請求項８～１６のうちのいずれか一項に記載のコンピュータ・プラットフォーム。

【請求項１９】

前記ソフトウェア実行プログラム（または前記ソフトウェア実行プログラムのうちの少なくとも１つ）は、前記トラステッド・モジュール(120)の公開鍵(228)と、各データに関するライセンスモデルとを含み、

前記オペレーティング・システムは、前記ソフトウェア実行プログラムに対し、該ソフトウェア実行プログラムの各データを使用すべきことを要求するように動作し、

前記ソフトウェア実行プログラムは、前記要求に応答して、前記プラットフォームまたは該プラットフォームのユーザが前記データを使用するための許諾を受けているか否かに

関するライセンス検査を、該ソフトウェア実行プログラムのライセンスモデルを用いて前記セキュア実行プログラム(204)に対して要求するように動作し、

前記セキュア実行プログラム(204)は、後者の要求に応答して、要求された前記ライセンス検査を実施し、該ライセンス検査の結果を前記トラステッド・モジュール(120)の秘密鍵(212)を用いて署名し、該署名された結果を前記ソフトウェア実行プログラムに返答するように動作し、

前記ソフトウェア実行プログラムは、前記返答に応答して、前記署名された結果の完全性を前記トラステッド・モジュール(120)の公開鍵(228)を用いて検査し、前記ライセンス検査の結果に関する完全性検査が成功すると、前記データの使用を前記オペレーティング・システムに対して要求するように動作する、請求項５、および請求項５に直接または間接に従属したときの請求項６～１８のうちのいずれか一項に記載のコンピュータ・プラットフォーム。

【請求項２０】

前記ソフトウェア実行プログラム（または前記ソフトウェア実行プログラムのうちの少なくとも１つ）は、前記トラステッド・モジュール(120)の公開鍵(228)と、各データに関するライセンスモデルとを含み、

前記オペレーティング・システムは、前記セキュア実行プログラム(204)に対し、特定のデータの使用を要求するように動作し、

前記セキュア実行プログラム(204)は、前記要求に応答して、前記トラステッド・モジュール(120)の秘密鍵(212)を用いて署名された前記特定のデータに関するライセンスモデルの要求を各ソフトウェア実行プログラムへ送信するように動作し、

前記ソフトウェア実行プログラムは、後者の要求に応答して、前記要求の完全性を前記トラステッド・モジュール(120)の前記公開鍵(228)を用いて検査し、該完全性の検査が成功すると、前記ライセンスモデルを前記セキュア実行プログラムへ送信するように動作し、

前記セキュア実行プログラム(204)は、前記ライセンスモデルを受信すると、該ライセンスモデルを用いてライセンス検査を実施し、該ライセンス検査が成功すると、前記データの使用を前記オペレーティング・システムに対して要求するように動作する、請求項５、および請求項５に直接または間接に従属したときの請求項６～１９のうちのいずれか一項に記載のコンピュータ・プラットフォーム。

【請求項２１】

前記セキュア実行プログラム(204)は少なくとも１つのライセンスモデルを含み、

前記オペレーティング・システムは、前記特定のデータの使用を前記セキュア実行プログラム(204)に対して要求するように動作し、

前記セキュア実行プログラム(204)は、前記要求に応答して、前記ライセンスモデルまたは前記ライセンスモデルのうちの１つを用いてライセンス検査を実施し、該ライセンス検査が成功すると、前記データの使用を前記オペレーティング・システムに対して要求するように動作する、請求項１～２０のうちのいずれか一項に記載のコンピュータ・プラットフォーム。

【請求項２２】

前記オペレーティング・システムは、前記セキュア実行プログラム(204)または前記ソフトウェア実行プログラムにのみ応答して前記特定のデータをインストールするようにプログラムされる、請求項１９～２１のうちのいずれか一項に記載のコンピュータ・プラットフォーム。

【請求項２３】

前記セキュア実行プログラム(204)は少なくとも１つのライセンスモデルを含み、

前記ソフトウェア実行プログラム（または前記ソフトウェア実行プログラムのうちの少なくとも１つ）は、前記トラステッド・モジュール(120)に対し、該トラステッド・モジュールの各データを使用すべきことを要求するように動作可能であり、

前記トラステッド・モジュール(120)内の前記セキュア実行プログラム(204)は、前記要

求に応答して、前記ライセンスモデル、または前記ライセンスモデルのうちの１つを用いてライセンス検査を実施し、該ライセンス検査が成功すると、前記データのインストールを前記オペレーティング・システムに対して要求するように動作する、請求項５に従属したときの請求項６、および請求項６に従属したときの請求項７～２２のうちのいずれか一項に記載のコンピュータ・プラットフォーム。

【請求項２４】

前記オペレーティング・システムは、前記トラステッド・モジュール(120)にのみ応答して前記特定のデータを使用するようにプログラムされる、請求項２３に記載のコンピュータ・プラットフォーム。

【請求項２５】

前記セキュア実行プログラム(204)からオペレーティング・システムへのデータ使用の要求は、前記専用通信路(122a)を介して供給される、請求項７に直接または間接に従属したときの請求項２０～２４のうちのいずれか一項に記載のコンピュータ・プラットフォーム。

【請求項２６】

前記トラステッド・モジュール(120)は、前記オペレーティング・システムに対するデータ使用の要求をログ記録するように動作する、請求項１９～２５のうちのいずれか一項に記載のコンピュータ・プラットフォーム。

【請求項２７】

ユーザ識別を含む着脱可能な他のトラステッド・モジュール(120)をさらに含み、

前記プラットフォームが、最初に述べたトラステッド・モジュール(120)と前記着脱可能なトラステッド・モジュール(19)との間で認証検査を実施するように動作し、

ライセンス検査の際に、前記セキュア実行プログラム(204)または前記ソフトウェア実行プログラムが、前記ユーザ識別を参照して前記ライセンス検査を実施するように動作する、請求項１９～２６のうちのいずれか一項に記載のコンピュータ・プラットフォーム。